# IMPLEMENTASI PENGAMANAN PESAN SMS DENGAN MANAJEMEN KUNCI PADA PONSEL BERBASIS ANDROID

# SKRIPSI

Diajukan untuk memenuhi persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh :

**ARNES DUSIMAR**

**NIM. 0610960009**

**KEMENTRIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**MALANG**

**2013**

**LEMBAR PERSETUJUAN**

IMPLEMENTASI PENGAMANAN PESAN SMS DENGAN MANAJEMEN KUNCI PADA PONSEL BERBASIS ANDROID

# SKRIPSI

Diajukan untuk memenuhi persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh :

**ARNES DUSIMAR**

**NIM. 0610960009**

Telah diperiksa dan disetujui oleh :

| Pembimbing I, | Pembimbing II, |
|---|---|
| **Issa Arwani, S.Kom., M.Sc.** | **Eko Sakti P, S.Kom., M.Kom.** |
| **NIP. 830922 06 11 0074** | **NIP.860805 06 11 0252** |

**LEMBAR PENGESAHAN**

IMPLEMENTASI PENGAMANAN PESAN SMS DENGAN MANAJEMEN KUNCI PADA PONSEL BERBASIS ANDROID

# SKRIPSI

Diajukan untuk memenuhi persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh :

**ARNES DUSIMAR**

**NIM. 0610960009**

Skripsi ini telah diuji dan dinyatakan lulus tanggal 29 Juli 2013

Penguji I

**Ismiarta Aknuranda, S.T.,M.Sc., Ph.D**
**NIP. 740719 06 11 0079**

Penguji II

**Aryo Pinandito, S.T., M.MT.**
**NIK. 870724 06 11 0374**

Penguji III

**Agi Putra Kharisma, S.T., M.T.**
**NIK. -**

Mengetahui
Ketua Program Studi Teknik Informatika

**Drs. Marji, MT**
**NIP. 19670801 199203 1 001**

# PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Juli 2013

Mahasiswa,

**Arnes Dusimar**

**NIM 0610960009**

# KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa yang telah melimpahkan segala berkat dan terang cahayanya sehingga Penulis dapat menyelesaikan skripsi dengan judul: " **Implementasi Pengamanan Pesan Sms Dengan Manajemen Kunci Pada Ponsel Berbasis Android**"

Skripsi ini diajukan sebagai syarat ujian akhir skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTIIK), Program Studi Informatika/Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Issa Arwani, S.Kom., M.Sc., selaku Dosen Pembimbing Skripsi I.
2. Eko Sakti Pramukantoro, S.Kom., M.Kom., selaku Dosen Pembimbing Skripsi II.
3. Drs. Marji, MT., selaku Dosen Penasehat Akademik sekaligus Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
7. Segenap staff dan karyawan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Orang tua Penulis dan saudara-saudaraku atas segala dukungan materi dan doa restunya kepada Penulis.
9. Rekan-rekan Ilkomers 2006 Program Studi Teknik Informatika yang telah memberikan dukungannya kepada penulis.
11. Sumpil, yang telah memberi motivasi dan inspirasi dalam menjalani hari-hari bersama skripsi yang telah disusun ini.

12. Anggota KDS 49 dan Tlogoagung 45X yang telah memberi semangat dan penghiburan disaat lelah.
13. Teman-teman KMK FEB UB dan KMK MIPA UB yang memberiku pengalaman rohani dan arti hidup dalam Kristus.
14. Semua pihak yang telah membantu terselesaikannya skripsi ini yang tidak dapat kami sebutkan satu per satu.

Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat Penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, 19 Juli 2013

Arnes Dusimar

Penulis

# ABSTRAK

Salah satu bentuk komunikasi yang sering dilakukan melalui telepon genggam adalah dengan menggunakan layanan pesan singkat atau yang biasa disebut SMS (*Short Message Service*). Layanan SMS sering dipergunakan karena layanan ini terdapat pada setiap ponsel dengan berbagai macam provider seluler.Selain itu, biaya yang murah juga menjadi alasan utama pengguna ponsel menggunakan layanan ini.

Terkadang ada kalanya informasi yang disampaikan melalui layanan SMS bersifat rahasia. Kenyataannya, saat ini layanan SMS pada dasarnya belum bisa menjamin kerahasiaan pesan. Dari kebutuhan akan kerahasiaan pesan, maka diperlukan suatu sistem yang dapat menjaga kerahasiaan isi pesan.

Kriptografi merupakan salah satu cara untuk menjaga kerahasiaan tersebut. Membangun perangkat lunak yang dapat mengenkripsi dan mendekripsi pesan diharapkan dapat mengatasi masalah diatas. Sistem manajemen kunci yang baik juga akan menambah keamanan dan kenyamanan pengguna layanan SMS dalam berkomunikasi. Dengan penerapan kriptografi dan sistem manajemen kunci diharapkan pengguna layanan SMS dapat mengamankan isi pesan yang dikirim.

Kata kunci : SMS, kriptografi, sistem manajemen kunci.

# ABSTRACT

*One way of communication that is often done via mobile phone is by using the short message service or commonly known as SMS (Short Message Service). SMS services are often used because this service are on every mobile phone with a variety of cellular provider. In addition, the low cost is also a major reason for the mobile users to use the service.*

*Sometimes, information is conveyed through SMS services are confidential. in reality, the current SMS service basically can not guarantee the confidentiality of the message. Of the need for confidentiality of a message, we need a system that can maintain the confidentiality of message content.*

*Cryptography is one way to maintain the secrecy. Build software that can encrypt and decrypt messages is expected to overcome the above problems. Good key management system will also increase the safety and convenience of SMS users to communicate.Implementation of cryptography and key management systems are expected to be able to secure the SMS service users content of messages sent.*

*Keywords : SMS, cryptography, key management system.*

# DAFTAR ISI


LEMBAR PERSETUJUAN ..... i
LEMBAR PENGESAHAN ..... ii
PERNYATAAN ORISINALITAS SKRIPSI ..... iii
KATA PENGANTAR ..... iv
ABSTRAK ..... vi
*ABSTRACT* ..... vii
DAFTAR ISI ..... viii
DAFTAR GAMBAR ..... xi
DAFTAR TABEL ..... xiii
BAB I PENDAHULUAN ..... 1
1.1. Latar Belakang ..... 1
1.2. Rumusan Masalah ..... 2
1.3. Batasan Masalah ..... 2
1.4. Tujuan ..... 2
1.5. Manfaat ..... 2
1.6. Sistematika Pembahasan ..... 2
BAB II KAJIAN PUSTAKA DAN DASAR TEORI ..... 4
2.1. Pengertian Kriptografi ..... 4
2.2. Fungsi Dasar Algoritma Kriptografi ..... 5
2.3. Macam-macam Algoritma Kriptografi ..... 5
2.3.1. Algoritma Simetri ..... 5
2.3.2. Algoritma Asimetri ..... 8
2.3.3. Fungsi *Hash* ..... 9
2.4. AlgoritmaTDES *(Triple Data Encryption Standart)* ..... 10
2.5. Algoritma AES ..... 20
2.6. Manajemen Kunci ..... 26
2.6.1. Pembangkitan Kunci ..... 27
2.6.2. Penyebaran Kunci ..... 28
2.6.3. Penyimpanan Kunci ..... 28
2.6.4. Penggunaan Kunci ..... 28

2.6.5. Penggunaan Kunci ..... 29
2.6.6. Penghancuran Kunci ..... 29
BAB III METODOLOGI PENELITIAN DAN PERANCANGAN ..... 30
3.1. Metode Penelitian ..... 30
3.2. Perancangan Sistem ..... 31
3.2.1. Batasan Perangkat Lunak ..... 31
3.2.2. Perancangan Perangkat Lunak ..... 32
3.2.2.1. Diagram *Use Case* ..... 32
3.2.2.2. *Sequence Diagram* ..... 33
3.2.3. Perancangan Algoritma Enkripsi dan Dekripsi ..... 36
3.2.3.1 Pembentukan Kunci TDES 192 Bit ..... 36
3.2.3.2 Enkripsi dengan TDES 192 Bit ..... 37
3.2.3.3 Pengiriman Kunci TDES melalui SMS ..... 37
3.2.3.4 Enkripsi *Session key* dengan AES 128 Bit ..... 39
3.2.3.5 Dekripsi dengan TDES 192 Bit ..... 39
3.2.3.6 Dekripsi dengan AES 128 bit ..... 40
3.2.3.7 Pertukaran Kunci dengan *Barcode Scanner* ..... 41
3.2.4. Perancangan Antar Muka ..... 42
3.2.4.1 Antar Muka Menu Utama ..... 42
3.2.4.2 Antar Muka Kirim Pesan ..... 43
3.2.4.3 Antar Muka Kotak Masuk ..... 44
3.2.4.4 Antar Muka Kotak Keluar ..... 44
3.2.4.5 Antar Muka Pengaturan Kunci ..... 45
BAB IV IMPLEMENTASI ..... 46
4.1. Lingkungan Implementasi ..... 46
4.1.1. Lingkungan Perangkat Keras ..... 46
4.1.2. Lingkungan Perangkat Lunak ..... 46
4.2 Implementasi Perangkat Lunak ..... 47
4.2.1. Instalasi Perangkat Lunak ..... 47
4.2.2. Menu *Log In* ..... 48
4.2.3. Antar Muka Menu Utama ..... 49
4.2.4. Pengaturan Kunci ..... 50

4.2.5. Tulis Pesan ..... 57
4.2.6. Pesan Masuk ..... 58
4.2.7. Pesan Keluar ..... 58
BAB V PENGUJIAN DAN ANALISIS ..... 60
5.1 Strategi Pengujian ..... 60
5.2 Hasil Pengujian ..... 66
5.2.1. Hasil Pengujian Fungsionalitas Perangkat Lunak ..... 66
5.2.2. Hasil Pengujian Pembengkakan Data *Ciphertext* ..... 73
5.3. Analisis Hasil Pengujian ..... 80
5.3.1. Analisis Hasil Uji Fungsionalitas Perangkat Lunak ..... 80
5.3.2. Analisis Hasil Uji Pembengkakan *Ciphertext* Perangkat Lunak . 80
5.4. Analisis Umum Hasil Uji ..... 82
BAB VI PENUTUP ..... 83
6.1. Kesimpulan ..... 83
6.2. Saran ..... 83
DAFTAR PUSTAKA ..... 84

# DAFTAR GAMBAR


Gambar 2.1 Kriptografi Kunci Simetrik ..... 6
Gambar 2.2 Skema Enkripsi dan Dekripsi chiper blok..... 8
Gambar 2.3 Kriptografi Kunci Asimetrik ..... 9
Gambar 2.4 Kriptografi Kunci Satu-Arah..... 10
Gambar 2.5 Skema Algoritma Kriptografi DES ..... 11
Gambar 2.6 Proses Komputasi Algoritma Kriptografi DES ..... 11
Gambar 2.7 Proses Pembangkitan Kunci-Kunci Internal DES..... 13
Gambar 2.8 Proses Subtitusi *bytes* ..... 20
Gambar 2.9 Proses penggeseran baris-baris *array* ..... 21
Gambar 2.10 Proses *mix colomns*..... 21
Gambar 2.11 Proses *add round key* ..... 22
Gambar 2.12 Diagram proses AES..... 22
Gambar 2.13 RootWord dan SubBytes dengan S-box ..... 23
Gambar 2.14 Penambahan dengan $W_{i-4}$ dan *Rcon*..... 24
Gambar 2.15 Penambahan dengan Wi-4..... 24
Gambar 2.16 Penambahan dengan $W_{i-4}$..... 25
Gambar 2.17 RootWord dan SubBytes dengan S-box ..... 25
Gambar 2.18 Penambahan dengan $W_{i-4}$ dan *Rcon*..... 26
Gambar 2.19 Hasil kunci putaran pada AES..... 26
Gambar 2.20 Daur Hidup Kunci ..... 27
Gambar 3.1 Tahapan-Tahapan Pembuatan Perangkat Lunak ..... 30
Gambar 3.2 Struktur *Ciphertext* Pertukaran Kunci dan *Ciphertext* Pesan..... 31
Gambar 3.3 Gambar diagram *use case* ..... 32
Gambar 3.4 Gambar *sequence diagram* kirim pesan..... 33
Gambar 3.5 Gambar *sequence diagram* terima pesan ..... 34
Gambar 3.6 Gambar *sequence diagram* pengaturan kunci ..... 35
Gambar 3.7 *Flowchart* Pembentukan Kunci..... 36

Gambar 3.8 *Flowchart* Enkripsi Pesan ..... 37
Gambar 3.9 *Flowchart* pengiriman *session key* melalui SMS ..... 38
Gambar 3.10 *Flowchart* Enkripsi *session key*. ..... 39
Gambar 3.11 *Flowchart* Dekripsi Pesan ..... 40
Gambar 3.12 *Flowchart* dekripsi *session key* ..... 41
Gambar 3.13 *Flowchart* dekripsi *session key* ..... 42
Gambar 3.14 Antar Muka Menu Utama ..... 43
Gambar 3.15 Antar Muka Tulis Pesan ..... 43
Gambar 3.16 Antar Muka Kotak Masuk ..... 44
Gambar 3.17 Antar Muka Pesan Kotak Keluar ..... 44
Gambar 3.18 Antar Muka Pengaturan Kunci ..... 45
Gambar 4.1 Antar Muka Proses Instalasi ..... 48
Gambar 4.2 Antar Muka Proses *Log in* ..... 49
Gambar 4.3 Antar Muka Menu Utama ..... 50
Gambar 4.4 Antar Muka Pengaturan Kunci ..... 51
Gambar 4.5 Antar Muka *Add Contact* ..... 52
Gambar 4.6 Antar Muka Pilih Aksi ..... 53
Gambar 4.7 Antar Muka Proses Membangkitkan Angka Acak ..... 54
Gambar 4.8 Antar Muka Pengaturan yang Telah Diperbaharui ..... 55
Gambar 4.9 Contoh Antar Muka Lihat QR ..... 56
Gambar 4.10 Antar Muka Pengambilan Kunci dengan *Barcode Scanner* ..... 57
Gambar 4.11 Antar Muka Buat Pesan ..... 57
Gambar 4.12 Antar Muka Pesan Masuk ..... 58
Gambar 4.13 Antar Muka Pesan Keluar ..... 59
Gambar 5.1 Grafik Perbandingan *Plaintext* dan *Ciphertext* ..... 81
Gambar 5.2 Grafik Persentase Pembengkakan *Ciphertext* ..... 81

# DAFTAR TABEL

Tabel 2.1 Tabel PC-1 ..... 12
Tabel 2.2 Tabel Pergeseran Bit Pada Tiap Putaran ..... 13
Tabel 2.3 Tabel PC-2 ..... 14
Tabel 2.4 Tabel IP ..... 15
Tabel 2.5 Tabel E ..... 15
Tabel 2.6 Tabel $S_n$ ..... 16
Tabel 2.7 Tabel Permutasi P ..... 18
Tabel 2.8 Tabel permutasi $IP^{-1}$ ..... 18
Tabel 2.9 Tabel S-Box ..... 21
Tabel 5.1 Tabel Data Kunci ..... 60
Tabel 5.2 Daftar Isi Pesan SMS pengujian fungsionalitas ..... 62
Tabel 5.3 Daftar Isi Pesan SMS pengujian pembengkakan data ..... 63
Tabel 5.4 Hasil Pengujian Fungsionalitas ..... 66
Tabel 5.5 Hasil Pengujian Pengiriman Dan Hasil *Ciphertext* Pesan ..... 69
Tabel 5.6 Hasil Pembacaan Pesan SMS dengan Kunci Asli ..... 70
Tabel 5.7 Hasil Pengujian Pembacaan Pesan SMS Dengan Kunci Palsu ..... 72
Tabel 5.8 Hasil Pengujian Pengiriman Dan Hasil *Ciphertext* Pesan ..... 73

# BAB I
# PENDAHULUAN

## 1.1. Latar Belakang

Telepon genggam merupakan salah satu benda yang tak dapat dipisahkan dari manusia. Saat ini telepon genggam sangat berkembang pesat seiring dengan kebutuhan komunikasi manusia. Selain itu, telepon genggam saat ini juga sudah dapat mengakomodasi banyak kebutuhan manusia antara lain internet, *games*, pemutar musik, GPS (*Global Positioning System*), *push email* dan *social media*. Fitur-fitur diatas dapat kita nikmati pada ponsel pintar dengan sistem operasi yang bermacam-macam. Salah satu sistem operasi yang banyak dipakai pada ponsel pintar saat ini adalah android.

Android merupakan sistem operasi berbasis Linux pada ponsel pintar dan komputer tablet. Saat ini android menjadi primadona sistem operasi pada telepon seluler. Android menyediakan *platform* terbuka sehingga para pengembang dapat berkreasi dalam membuat aplikasi. Selain harga yang terjangkau dan tampilan antarmuka yang menarik, android juga banyak menyediakan aplikasi berkualitas yang selalu terkini.

Salah satu bentuk komunikasi yang sering dilakukan melalui telepon genggam adalah dengan menggunakan layanan pesan singkat atau yang biasa disebut SMS (*Short Message Service*). Terkadang ada kalanya informasi yang kita kirim melalui SMS bersifat rahasia. Staelium UK, salah satu operator seluler di Inggris, mengeluarkan satu layanan bernama "*stealth text*". Cara kerja layanan ini adalah menghapus pesan segera setelah 40 detik pesan dibaca. Alternatif lain untuk mengamankan data pesan singkat adalah kriptografi.

Kriptografi dapat diartikan sebagai teknik-teknik penyandian data. Kriptografi terbagi menjadi 2 proses utama yaitu enkripsi dan dekripsi. Enkripsi proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Sedangkan dekripsi adalah proses pengembalian data hasil enkripsi.

Selain kriptografi, sistem manajemen kunci juga diperlukan agar transmisi data menjadi aman. Salah satu implementasi sistem manajemen kunci yang efektif adalah dengan membuat kunci sementara atau *session key*. Kunci ini adalah dihasilkan dengan membuat angka acak.

## 1.2. Rumusan Masalah

Dari latar belakang yang telah dipaparkan diatas, adapun rumusan masalah dari skripsi ini adalah bagaimana merancang dan mengimplementasikan teknologi enkripsi, dekripsi, dan sistem manajemen kunci kriptografi pesan SMS pada telepon genggam berbasis android.

## 1.3. Batasan Masalah

Adapun batasan permasalahan pada skripsi ini adalah:

1. Perangkat lunak yang dibangun pada android dengan sistem operasi versi 2.3 (*Gingerbread*).
2. Telepon genggam dari pengirim dan penerima pesan singkat telah dipasang aplikasi ini.
3. Telepon genggam dari pengirim dan penerima pesan singkat telah dipasang aplikasi *WhatsApp* dan *Barcode scanner*.

## 1.4. Tujuan

Tujuan dari skripsi ini adalah meningkatkan kenyamanan dan keamanan pengguna layanan pesan singkat dengan cara membangun perangkat lunak yang dapat menjaga kerahasiaan pesan yang dikirim melalui fasilitas SMS.

## 1.5. Manfaat

Dari skripsi ini diharapkan agar pengguna layanan pesan singkat dapat mengirim pesan yang bersifat rahasia dengan aman.

## 1.6. Sistematika Pembahasan

Adapun enam sistematika pembahasan dari skripsi ini secara garis besar adalah :

**Bab I: Pendahuluan**

Dalam bab ini penulis menguraikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.

**Bab II: Kajian Pustaka dan Dasar Teori**

Pada bab ini dibahas mengenai teori-teori yang mendukung pembahasan bab selanjutnya.

**Bab III: Metodologi Penelitian dan Perancangan**

Bab ini berisi metode atau langkah-langkah yang dilakukan dalam penelitian. Selain itu penulis juga membahas tentang perancangan sistem enkripsi dan dekripsi file, gambaran umum rancangan aplikasi enkripsi, perancangan dan *user interface* aplikasi enkripsi.

**Bab IV: Implementasi**

Dalam bab ini Penulis akan membahas tentang implementasi algoritma perangkat lunak yang dibuat. Disertai juga gambaran antarmuka dari perangkat lunak yang dibangun.

**Bab V: Pengujian dan Analisis**

Bab ini berisi teknik pengujian, hasil pengujian dan analisis hasil pengujian dari perangkat lunak yang telah dibangun.

**Bab VI: Penutup**

Merupakan rangkuman dari laporan skripsi.

# BAB II

# KAJIAN PUSTAKA DAN DASAR TEORI

## 2.1. Pengertian Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: ”*cryptos*” artinya ”*secret*” (rahasia), sedangkan ”*graphein*” artinya ”*writing*” (tulisan). Jadi, kriptografi berarti ”*secret writing*” (tulisan rahasia) [MUN-06].

Kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu “*kriptos*” yang berarti tersembunyi/disembunyikan dan “*graphia*” yang berarti tulisan. Dengan demikian, istilah kriptografi dapat diartikan seni dan ilmu membuat sandi. Disisi lain, istilah kriptanalisis dapat diartikan seni dan ilmu membongkar sandi [PAU-98]. Pokok pembahasan kriptologi meliputi kriptografi dan kriptanalisis.

Kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untukmendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar. Jika suatu teks-kode berhasil diubah menjadi teks-asli tanpa menggunakan kunci yang sah, proses tersebut dinamakan *breaking code*. Hal ini dilakukan oleh para kriptanalis. Analisis kode juga dapat menemukan kelemahan dari suatu algoritma kriptografi dan akhirnya dapat menemukan kunci atau teks-asli dari teks-kode yang dienkripsi dengan algoritma tertentu [ARI-08].

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun, kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah disandi.
2. 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, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. 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 datanya, waktu pengiriman, dan lain-lain.
4. Non-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

## 2.2. Fungsi Dasar Algoritma Kriptografi

Fungsi-fungsi yang mendasar dalam kriptografi adalah enkripsi dan dekripsi. Enkripsi adalah proses mengubah suatu pesan asli (*plaintext*) menjadi suatu pesan dalam bahasa sandi (*ciphertext*). Sedangkan dekripsi adalah proses mengubah pesan dalam suatu bahasa sandi menjadi pesan asli kembali.

$$C = E(M) \quad (2\text{-}1)$$

$$M = D(C) \quad (2\text{-}2)$$

Dimana :

E = proses enkripsi

D = proses dekripsi

M = pesan asli

C = pesan dalam bahasa sandi (untuk ringkasnya disebut sandi) [HER-99]

## 2.3. Macam-macam Algoritma Kriptografi

Berdasarkan kunci yang digunakan, secara umum kriptografi dibagi menjadi 3, yaitu:

### 2.3.1. Algoritma Simetri

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi maupun dekripsi. Algoritma ini sudah ada sejak lebih dari 4000 tahun yang lalu. Sebelum komputer ada, kriptografi dilakukan dengan algoritma berbasis karakter. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan

tersebut agar bisa mendekripsikan pesan yang terkirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan.

Kelebihan kriptografi kunci simetrik adalah:

1. Kriptografi kunci simetri membutuhkan waktu yang singkat untuk melakukan proses enkripsi atau dekripsi.
2. Ukuran kunci simetri relatif lebih pendek.
3. Otentikasi pengiriman pesan langsung diketahui dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima saja.

Sedangkan kelemahan kriptografi kunci simetrik adalah:

1. Kunci simetri harus dikirim melalui saluran komunikasi yang aman, dan kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci.
2. Kunci harus sering diubah setiap kali melakukan komunikasi.

Gambar 2.1 Kriptografi Kunci Simetrik

Algoritma yang memakai kunci simetri di antaranya adalah :

1. *Data Encryption Standard (DES)*
2. *RC2, RC4, RC5, RC 6*
3. *International Data Encryption Algorithm (IDEA)*
4. *Advanced Encryption Standard (AES)*
5. *On Time Pad (OTP)*
6. *A5*

Algoritma kriptografi simetri dapat dibagi menjadi dua kategori, yaitu:

1. Cipher aliran (*stream cipher*)

   Algoritma kriptografi ini beroperasi pada *plaintext*/*ciphertext* dalam bentuk bit tunggal. Rangkaian bit dienkripsikan/didekripsikan bit perbit.

2. Cipher blok (*block cipher*)

   Algoritma kriptografi beroperasi pada *plaintext*/*ciphertext* dalam bentuk blok bit. Rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya [MUN-04]. *Plaintext* dibagi menjadi blok-blok bit dengan panjang yang sama, biasanya 64 bit, 128 bit dan 256 bit. Enkripsi dilakukan terhadap blok bit *plaintext* menggunakan bit-bit kunci (yang ukurannya sama dengan blok *plaintext*). Algoritma enkripsi menghasilkan blok *ciphertext* yang berukuransama dengan blok *plaintext*. Dekripsi dilakukan dengan cara yang sama seperti enkripsi.

   Misalkan blok *plaintext* (P) yang berukuran mbit dinyatakan sebagai vektor.

   $$P = (p1, p2, ..., pm) \quad (2\text{-}3)$$

   yang dalam hal ini pi adalah bit 0 atau bit 1 untuk

   $$i = 1, 2, \ldots, m, \quad (2\text{-}4)$$

   dan blok *ciphertext* (C) adalah

   $$C = (c1, c2, ..., cm) \quad (2\text{-}5)$$

   yang dalam hal ini ci adalah bit 0 atau bit 1 untuk

   $$i = 1, 2, \ldots, m. \quad (2\text{-}6)$$

   Bila *plaintext* dibagi menjadi n buah blok,barisan blok-blok *plaintext* dinyatakan sebagai

   $$(P1, P2, \ldots, Pn) \quad (2\text{-}7)$$

   Untuk setiap blok *plaintext* Pi, bit-bit penyusunnya dapat dinyatakan sebagai vektor

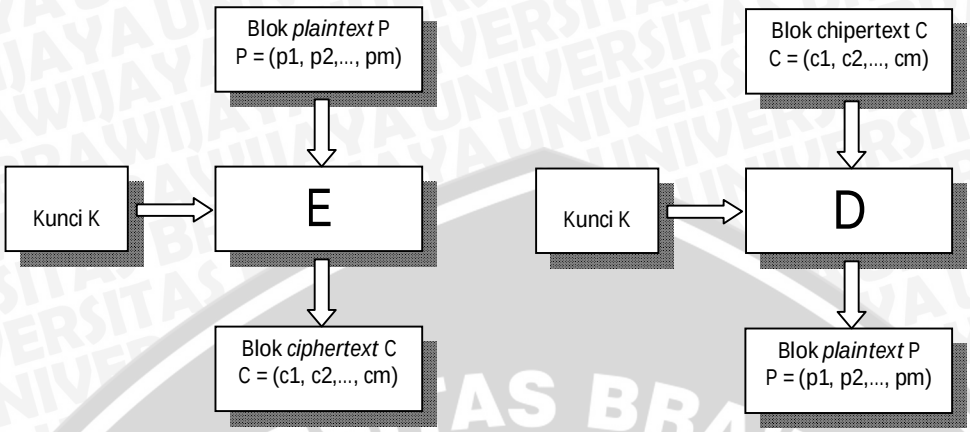
   $$Pi = (pi1, pi2, ..., pim) \quad (2\text{-}8)$$

   Enkripsi dengan kunci K dinyatakan dengan persamaan

   $$Ek(P) = C, \quad (2\text{-}9)$$

   sedangkan dekripsi dengan kunci K dinyatakan dengan persamaan

   $$Dk(C) = P \quad (2\text{-}10)$$




Gambar 2.2 Skema Enkripsi dan Dekripsi chiper blok

### 2.3.2. Algoritma Asimetri

Algoritma asimetri adalah kriptografi yang menggunakan kunci yang berbeda untuk melakukan proses enkripsi dan dekripsi, terkenal dengan istilah '*Public Key Cryptography*'.Kriptografi ini memiliki dua kunci (*e*,*d*). Kunci *e* dibuat untuk umum sehingga disebut kunci publik, sementara kunci *d* tetap disimpan aman dan rahasia sehingga disebut kunci privat. Demi keamanan maka harusnya menghitung *d* menjadi lebih sulit bila *e* diketahui [MEN-97].

Kriptografi kunci publik dapat dianalogikan seperti kotak surat yang terkunci dan memiliki lubang untuk memasukkan surat. Kotak surat diletakkan di depan rumah pemiliknya. Setiap orang dapat memasukkan surat ke dalam kotak tersebut, tetapi hanya pemilik kotak yang dapat membuka kotak dan membaca suratnya karena hanya ia yang memiliki kuncinya [RAT-06].

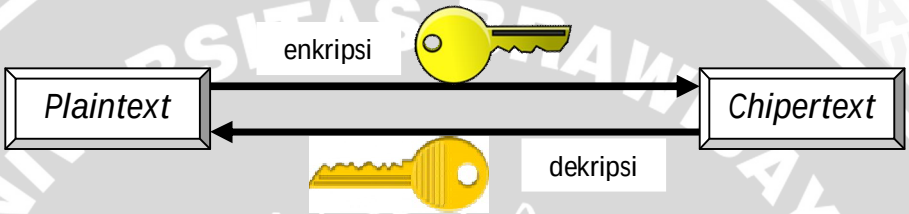
Kunci-kunci tersebut berhubungan satu sama lain. Dengan kunci publik orang dapat mengenkripsi pesan tetapi tidak bisa mendekripsikannya. Hanya orang yang memiliki kunci rahasia yang dapat mendekripsikan pesan tersebut. Algoritma asimetri bisa mengirimkan pesan dengan lebih aman daripada algoritma simetri.

Kelebihan kriptografi kunci asimetrik adalah:

1. Hanya kunci privat yang perlu dijaga kerahasiaannya.
2. Dapat digunakan dalam pengamanan pengiriman kunci simetri.
3. Sebagian algoritma dapat menambahkan otentikasi pesan.

Sedangkan kelemahan kriptografi kunci asimetrik adalah:

1. Ukuran kunci relatif lebih besar daripada kunci simetri.
2. Kriptografi kunci asimetri membutuhkan waktu yang relatif lebih lama untuk melakukan proses enkripsi atau dekripsi.




Gambar 2.3 Kriptografi Kunci Asimetrik

Algoritma yang memakai kunci public di antaranya adalah :

*a) Digital Signature Algorithm (DSA)*
*b) RSA*
c) Diffle-Hellman (DH)
*d) Elliptic Curve Cryptography (ECC)*
e) Kriptografi Quantum

### 2.3.3. Fungsi *Hash*

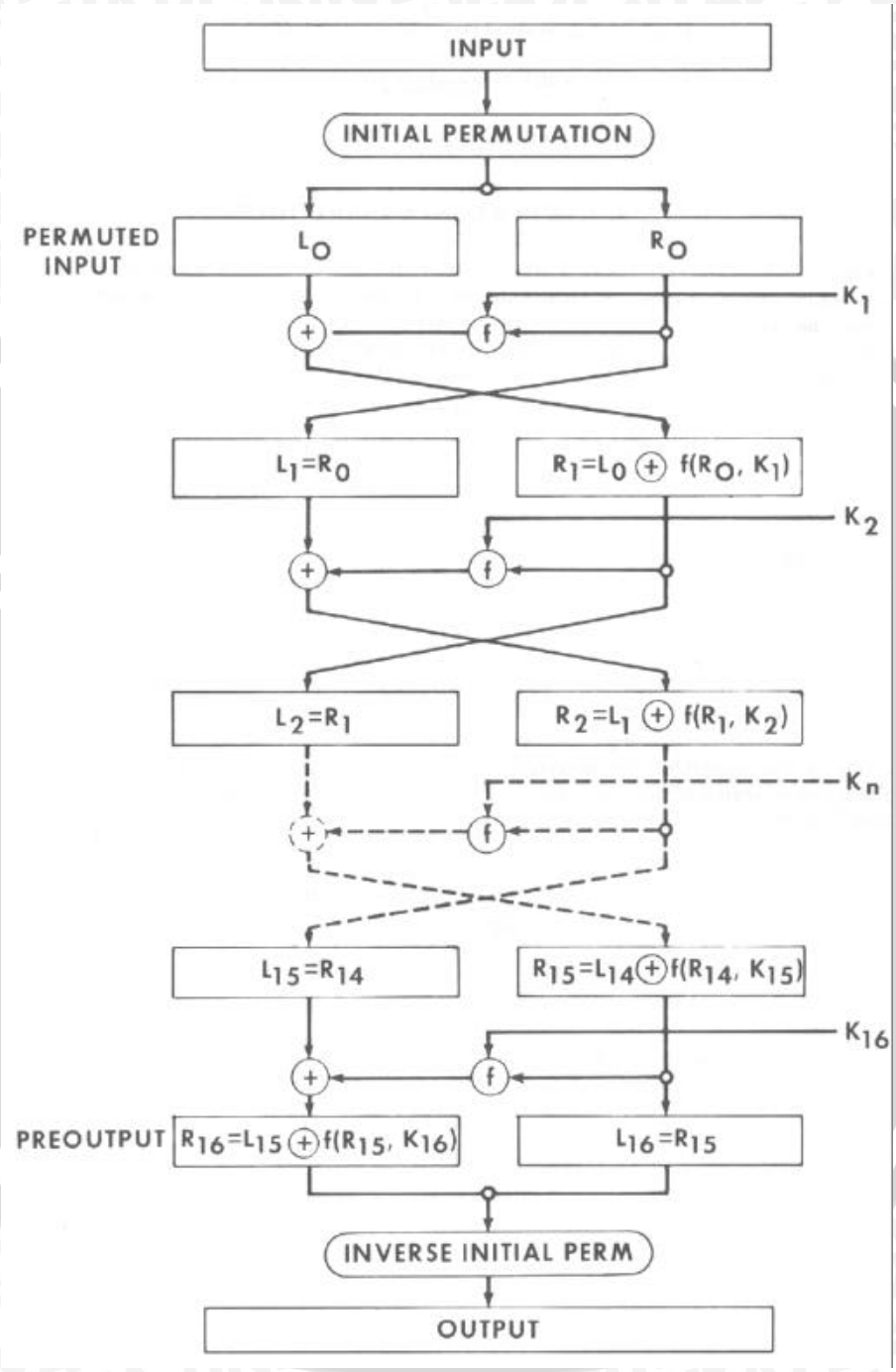
Fungsi *Hash* sering disebut dengan funsi satu arah (*one-way function*), *message digest*, *fingerprint,* fungsi kompresi dan *message authentication code (MAC)*, merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. Fungsi *Hash* biasanya diperlukan bila ingin membuat sidik jari dari suatu pesan. Sidik jari pada pesan merupakan suatu tanda bahwa pesan tersebut benar-benar berasal dari orang-orang yang diinginkan.

Gambar 2.4 Kriptografi Kunci Satu-Arah

## 2.4. AlgoritmaTDES *(Triple Data Encryption Standart)*

Algoritma kriptografi TDES *(Triple Data Encryption Standart)* adalah pengembangan dari algoritma kriptografi DES *(Data Encryption Standart)*. DES merupakan salah satu algoritma yang dikembangkan oleh NBS (*National Bureau of Standards* pada 15 Juli 1977. Algoritma DES mengadopsi dan memodifikasi algoritma Lucifer dengan bantuan NSA (*National Security Agency*). Pada awalnya Lucifer beroperasi dengan kunci 128 bit dan blok *input* 64 bit. Sedangkan DES beroperasi dengan kunci 64 bit (efektif 56 bit) dan blok *input* 64 bit.

Gambar 2.5 Skema Algoritma Kriptografi DES



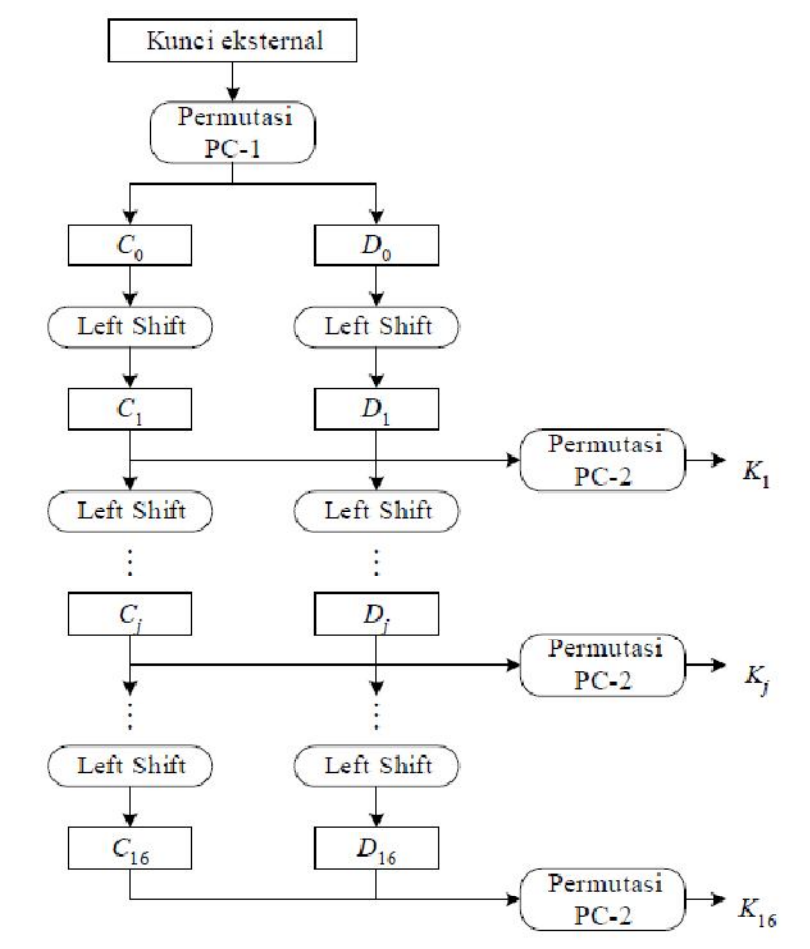

Gambar 2.6 Proses Komputasi Algoritma Kriptografi DES

Langkah awal enkripsi DES adalah siapkan kunci untuk melakukan enkripsi ditiap putarannya. Kunci 64-bit dipermutasi sesuai dengan tabel PC-1. Angka pertama dari tabel adalah "57", ini berarti bahwa bit-57 dari kunci K asli menjadi bit pertama dari permutasi kunci K+. Bit ke-49 bit kunci asli menjadi bit kedua dari kunci permutasi. Bit ke-4 bit dari kunci asli adalah bit terakhir kunci permutasi dengan catatan hanya 56 bit dari kunci asli muncul di kunci permutasi.

Tabel 2.1 Tabel PC-1

| | | | | | | |
|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Selanjutnya, bagi kunci menjadi bagian kiri dan bagian kanan, $C_0$ dan $D_0$, di mana masing-masing setengah memiliki 28 bit.Dengan $C_0$ dan $D_0$yang sudah terdefinisi, sekarang buat enam belas blok $C_n$ dan $D_n$. Setiap pasang blok $C_n$ dan $D_n$ dibentuk dari pasangan sebelumnya masing $C_{n-1}$ dan $D_{n-1}$, untuk n = 1, 2, ..., 16, dengan menggunakan jadwal pergeseran dari blok sebelumnya. Untuk melakukan pergeseran kiri, pindah setiap bit satu tempat ke kiri, kecuali untuk bit pertama [NIS-99].




Gambar 2.7 Proses Pembangkitan Kunci-Kunci Internal DES

Tabel 2.2 Tabel Pergeseran Bit Pada Tiap Putaran

| Iterasi ke- | Jumlah pergeseran kiri |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 2 |
| 6 | 2 |

| | |
|---|---|
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 2 |
| 11 | 2 |
| 12 | 2 |
| 13 | 2 |
| 14 | 2 |
| 15 | 2 |
| 16 | 1 |

Sekarang bentuk kunci $K_n$, untuk n dari 1 sampai 16, dengan menerapkan tabel permutasi berikut untuk masing-masing pasangan$C_nD_n$. Setiap pasangan memiliki 56 bit, tapi PC-2 hanya menggunakan 48 ini.Bit pertama dari Kn adalah bitke-14$C_nD_n$, bit kedua dari Kn adalah bit ke-17 $C_nD_n$, dan seterusnya, diakhiri dengan bitke-48 dari $K_n$ menjadi bit ke-32 dari $C_nD_n$.

Tabel 2.3 Tabel PC-2

| | | | | | |
|---|---|---|---|---|---|
| 14 | 17 | 11 | 24 | 1 | 5 |
| 3 | 28 | 15 | 6 | 21 | 10 |
| 23 | 19 | 12 | 4 | 26 | 8 |
| 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 |
| 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 |
| 46 | 42 | 50 | 36 | 29 | 32 |

Setelah mendapatkan kunci untuk 16 putaran DES, lakukan initial permutation (IP), yaitu permutasi blok *plaintext* sepanjang 64 bit yang akan dienkripsi. Dimana masukan yang sudah dipermutasi ditata sesuai dengan tabel dibawah ini. Bit ke-58 dari blok *plaintext* menjadi bit pertama dari IP, bit ke-50

dari blok *plaintext* menjadi bit kedua dari IP dan seterusnya hingga bit ke-7 dari blok *plaintext* menjadi bit terakhir dari IP.

Tabel 2.4 Tabel IP

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 20 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Sekarang kita lanjutkan melalui 16 iterasi (16 ronde) menggunakan fungsi f yang beroperasi pada dua blok data 32 bit (blok Ldan blok R) dan Kkunci 48 bit. Hasil operasi f menghasilkan blok L dan blok R yang baru. Dan + menunjukkan operasi XOR. Untuk ndari 1 sampai 16 menghasilkan blok akhir $L_{16}R_{16}$ dengan melakukan proses :

$$L_n = R_{n-1} \quad (2\text{-}11)$$

$$R_n = L_{n-1} + f\,(R_{n-1}, K_n) \quad (2\text{-}12)$$

Untuk menghitung f, pertama kita memperluas setiap blok $R_{n-1}$ dari 32 bit menjadi 48 bit. Hal ini dilakukan dengan menggunakan tabel seleksi yang mengulangi beberapa bit dalam $R_{n-1}$. Dengan menggunakan tabel E dibawah ini, maka didapatkan E ($R_{n-1}$) memiliki blok *input* 32 bit dan 48 bit *output* blok.Biarkan E sedemikian rupa sehingga 48 bit *output,* ditulis sebagai 8 blok dari 6 bit masing-masing, diperoleh dengan memilih bit masukan dalam rangka sesuai dengan tabel berikut:

Tabel 2.5 Tabel E

| | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

Selanjutnya dalam perhitungan f, kita XOR *output* E ($R_{n-1}$) dengan $K_n$ kunci:

$$E\ K_n + (R_{n-1}) \quad (2\text{-}13)$$

Setelah memperluas $R_{n-1}$ dari 32 bit ke 48 bit, menggunakan tabel seleksi, dan XOR hasil dengan $K_n$ kunci. Setelah memiliki 48 bit, atau delapan kelompok 6 bit. 6 bit digunakan sebagai alamat dalam tabel S-box. Setiap kelompok terdiri dari  bit akan memberi kita sebuah alamat di S-box yang berbeda. Alamat itu akan menjadi 4 bit. 4 bit iniakan menggantikan 6 bit tadi. Hasil akhirnya adalah bahwa delapan kelompok 6 bit diubah menjadi delapan kelompok dari 4 bit (*output* 4-bit dari S-box) untuk 32 bit secara total. Hasil sebelumnya, yaitu 48 bit, berupa :

$$E\ K_n + (R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8 \quad (2\text{-}14)$$

di mana setiap Bi adalah kelompok enam bit, kemudian hitung :

$$S_1\ (B_1)\ S_2\ (B_2)\ S_3\ (B_3)\ S_4\ (B_4)\ S_5\ (B_5)\ S_6\ (B_6)\ S_7\ (B_7)\ S_8\ (B_8) \quad (2\text{-}15)$$

Dengan tabel $S_n$ sebagai berikut :

Tabel 2.6 Tabel $S_n$

$S_1$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

$S_2$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |

$S_3$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |

$S_4$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |

$S_5$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 16 | 13 | 0 | 14 | 9 |
| 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |

$S_6$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |

$S_7$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |

$S_8$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

Tahap akhir dalam perhitungan f adalah untuk melakukan permutasi P dari *output* S-box untuk mendapatkan nilai akhir f:

$$f = P\,(S_1(B_1)\,S_2(B_2)...S_8(B_8)) \qquad (2\text{-}16)$$

Permutasi P didefinisikan dalam tabel berikut. P menghasilkan *output* 32-bit dari masukan 32-bit dengan permutasi bit dari blok *input*.

Tabel 2.7 Tabel Permutasi P

| | | | |
|---|---|---|---|
| 16 | 7 | 20 | 21 |
| 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 |
| 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 |
| 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 |
| 22 | 11 | 4 | 25 |

Di putaran selanjutnya, kita akan memiliki $L_n = R_{n-1}$, dan kemudian kita harus menghitung $R_n = L_{n-1} + f(R_{n-1}, K_n)$, dan seterusnya untuk 16 putaran. Pada akhir putaran ke-16dihasilkan blok $L_{16}$ dan $R_{16}$, kemudian balikkan urutan dua blok dan dihasilkan blok 64-bit$R_{16}L_{16}$. Langkah terakhir adalah menerapkan permutasi akhir $IP^{-1}$ seperti yang didefinisikan oleh tabel berikut:

Tabel 2.8 Tabel permutasi $IP^{-1}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

Proses dekripsi terhadap *ciphertext* merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K1, K2, …, K16, maka pada proses dekripsi urutan kunci yang digunakan adalah K16, K15, …, K1.Untuk tiap putaran 16, 15, …, 1, keluaran pada setiap putaran *deciphering* adalah :

$$L_i = R_{i-1} \quad (2\text{-}17)$$

$$R_i = L_{i-1} + f(R_{i-1}, K_i) \quad (2\text{-}18)$$

yang dalam hal ini, ($R_{16}$, $L_{16}$) adalah blok masukan awal untuk dekripsi. Blok ($R_{16}$, $L_{16}$) diperoleh dengan mempermutasikan *ciphertext* dengan matriks

permutasi IP-1. Keluaran dari proses ini adalah ($L_0$, $R_0$). Dengan permutasi awal IP akan didapatkan kembali blok *plaintext* semula.

Setelah itu, tinjau kembali proses pembangkitan kunci internal pada Gambar 2.7. Selama proses dekripsi, $K_{16}$ dihasilkan dari ($C_{16}$, $D_{16}$) dengan permutasi PC-2. Tentu saja ($C_{16}$, $D_{16}$) tidak dapat diperoleh langsung pada awal proses dekripsi. Tetapi karena ($C_{16}$, $D_{16}$) = ($C_0$, $D_0$), maka $K_{16}$ dapat dihasilkan dari ($C_0$, $D_0$) tanpa perlu lagi melakukan pergeseran bit. ($C_0$, $D_0$) yang merupakan bit-bit dari kunci eksternal K yang diberikan pengguna pada waktu dekripsi.

Selanjutnya, $K_{15}$ dihasilkan dari ($C_{15}$, $D_{15}$) yang mana ($C_{15}$, $D_{15}$) diperoleh dengan menggeser $C_{16}$ (yang sama dengan $C_0$) dan $D_{16}$ (yang sama dengan $C_0$) satu bit ke kanan. Sisanya, $K_{14}$ sampai $K_1$ dihasilkan dari ($C_{14}$, $D_{14}$) sampai ($C_1$, $D_1$). Dengan catatan bahwa ($C_{i-1}$, $D_{i-1}$) diperoleh dengan menggeser $C_i$ dan $D_i$ dengan cara yang sama seperti pada Tabel 2.2, tetapi pergeseran kiri (*left shift*) diganti menjadi pergeseran kanan (*right shift*).

Saat ini DES dianggap sudah diaggap kurang aman karena kunci yang digunakan terlalu pendek yaitu 56 bit. Karena itulah algoritma ini dikembangkan lagi menghasilkan algoritma TDES (Triple Data Encryption Standart). Algoritma TDES adalah penerapan DES dengan dua kunci (112 bit) atau tiga kunci (168 bit). Kemudian hasilnya didekripsi dengan menggunakan kunci kedua. Kemudian langkah terakhir adalah hasil dekripsi dengan kunci kedua tadi dienkripsi kembali dengan kunci ketiga. Dalam proses ini terdapat tiga langkah DES sehingga disebut *triple-DES*. Untuk TDES dengan dua kunci maka kunci ketiga sama dengan kunci pertama.

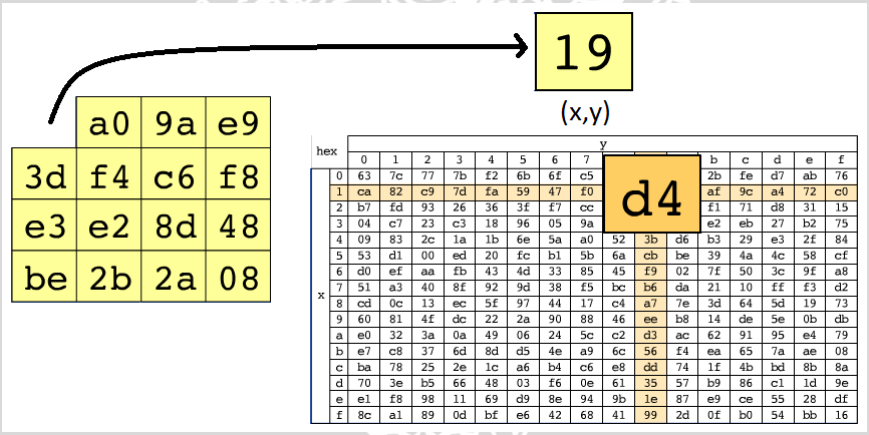
Pengolahan dasar yang terlibat dalam TDES adalah sebagai berikut: Sebuah blok input dibaca ke dalam DES pertama (DES1) dan dienkripsi menggunakan kunci pertama (*key*1). Output yang dihasilkan dari tahap ini dibaca langsung ke DES kedua (DES2) dan didekripsi menggunakan kunci kedua (*key*2). Output yang dihasilkan oleh tahap kedua langsung membaca ke DES ketiga (DES3) dan dienkripsi menggunakan kunci ketiga (*key*3). Yang dihasilkan blok output yang digunakan, sesuai dengan mode dilaksanakan, dalam perhitungan *ciphertext* [KEL-12].

## 2.5. Algoritma AES

*Advanced Encryption Standard* (AES) merupakan algoritma kriptografi yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok *ciphertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*; sebaliknya dekripsi adalah merubah *ciphertextt* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES ini mengunakan kunci kriptografi 128, 192 dan 256 bit untuk mengenkripsi dan dekripsi data pada blok 128 bit [ARI-11].

Garis besar proses pada algoritma ini adalah :

1. *AddRoundKey* : melakukan XOR antara *state* awal (*plaintext*) dengan *chiper key*. Tahap ini disebutjuga *initial round*.
2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah :
   a. *SubBytes* : subtitusi *byte* dengan menggunakan tabel subtitusi (S-box)

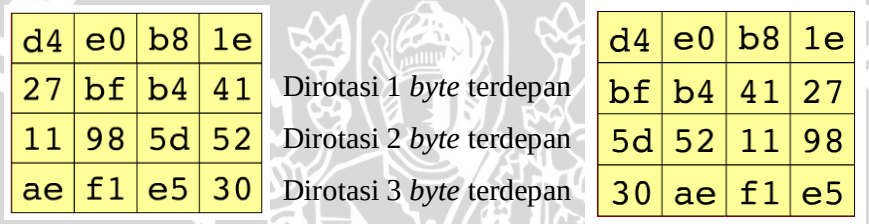



Gambar 2.8 Proses Subtitusi *bytes*

Tabel 2.9 Tabel S-Box

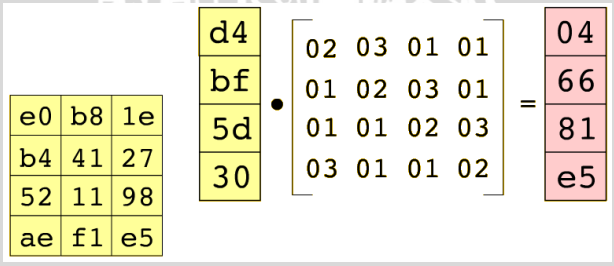
| | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xa | xb | xc | xd | xe | xf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| 1x | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 2x | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 3x | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 4x | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 5x | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 6x | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| 7x | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 8x | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 9x | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| ax | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| bx | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| cx | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| dx | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| ex | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| fx | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

b. *ShiftRows* : pergeseran baris-baris array *state* secara *wrapping*.



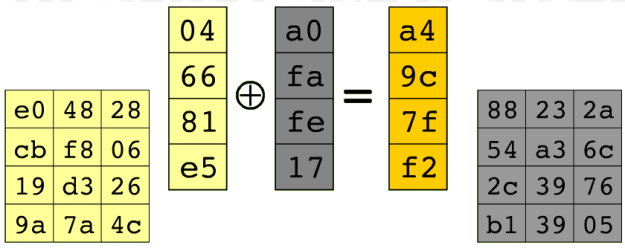

Gambar 2.9 Proses penggeseran baris-baris *array*

c. *MixColumns* : mengacak data di masing-masing kolom *arraystate.*




Gambar 2.10 Proses *mix colomns*

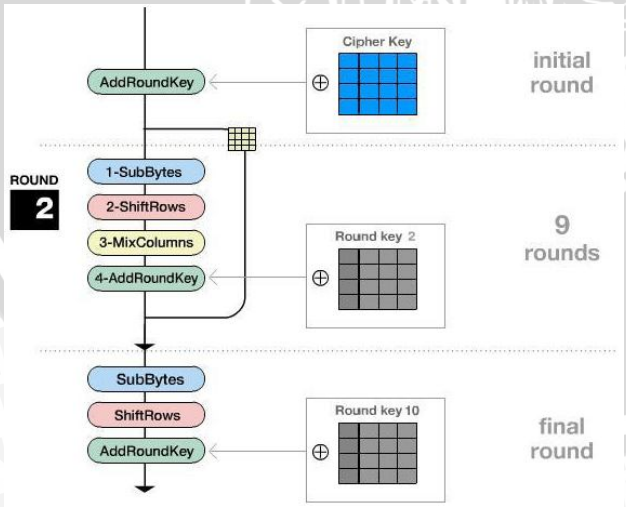
d. *AddRoundKey* : melakukan XOR antara *state* dan*round key*.




Gambar 2.11 Proses *add round key*

3. *Final round* : proses untuk putaran terakhir.
   *a. SubBytes*
   *b. ShiftRows*
   *c. AddRoundKey*

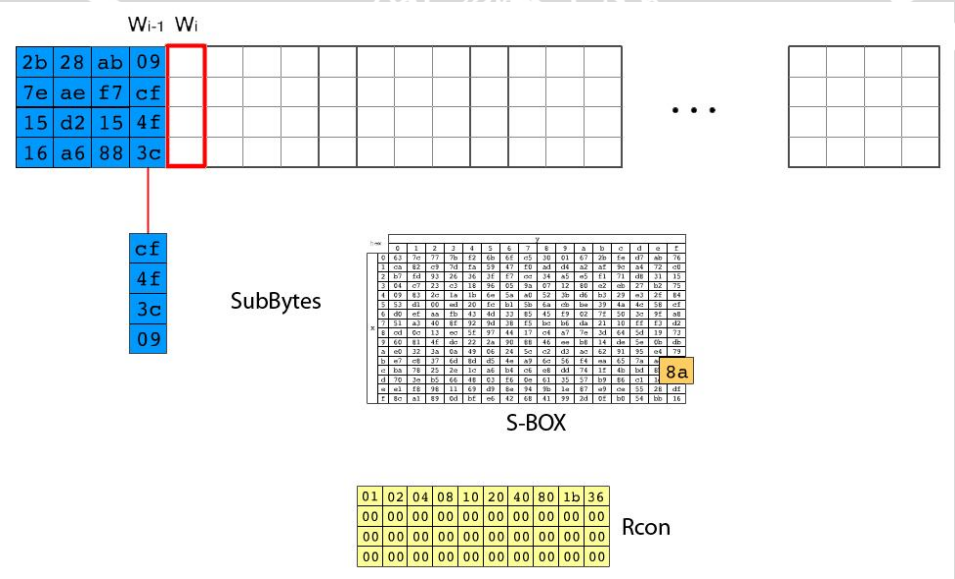
Algoritma AES terdiri dari sepuluh putaran enkripsi, seperti dapat dilihat pada Gambar 2.12. Pertama kunci 128-bit diperluas menjadi sebelas disebut kunci bulat, masing-masing 128 bit dalam ukuran. Setiap putaran meliputi transformasi menggunakan kunci cipher yang sesuai untuk menjamin keamanan enkripsi [KRE-09].



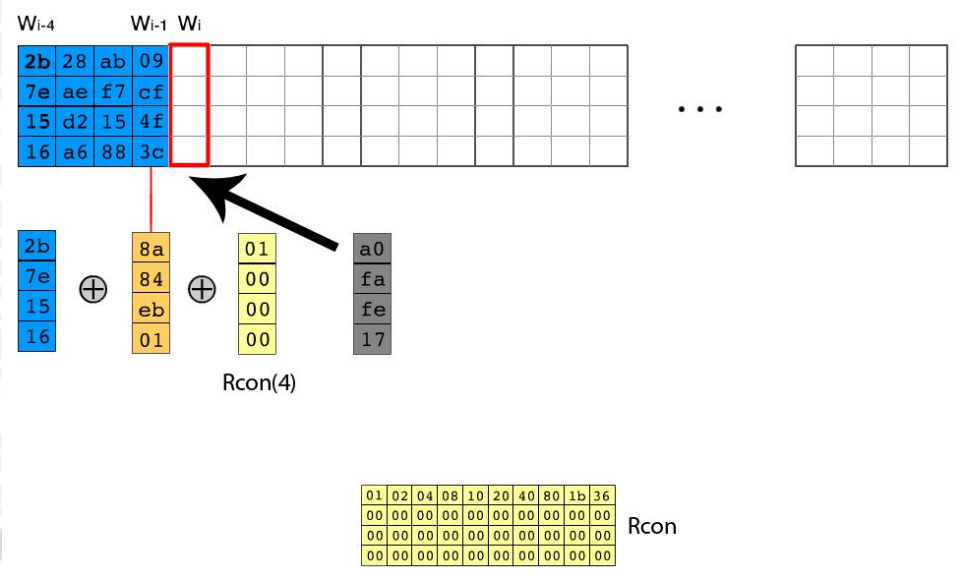

Gambar 2.12 Diagram proses AES

Sedangkan untuk pembangkitan kunci putaran dilakukan proses sebagai berikut :

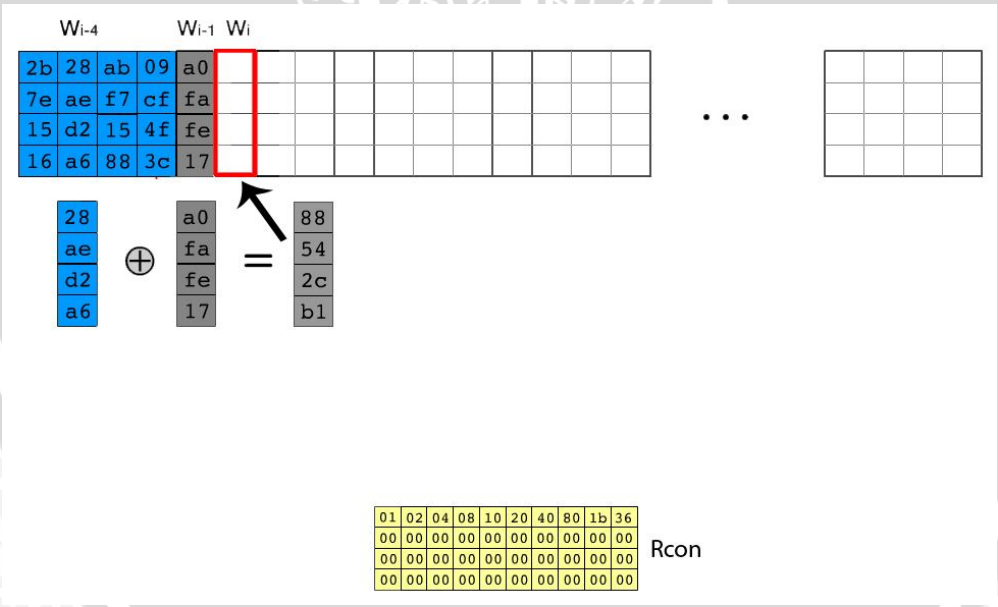
1. *Rotword* kolom pertama.
2. *SubBytes* dengan tabel S-box.
3. Penambahan dengan $W_{i-4}$ dan *Rcon* dan hasilnya diisi pada kolom 1 kunci putaran ke-2.
4. Untuk mengisi kolom 2, kunci putaran ke-2 dilakukan penambahan dengan $W_{i-4}$.
5. Untuk mengisi kolom 3, kunci putaran ke-2 dilakukan penambahan dengan $W_{i-4}$.
6. Untuk mengisi kolom 4, kunci putaran ke-2 dilakukan penambahan dengan $W_{i-4}$.
7. Langkah diatas diulangi terus sampai didapat semua kunci putaran.



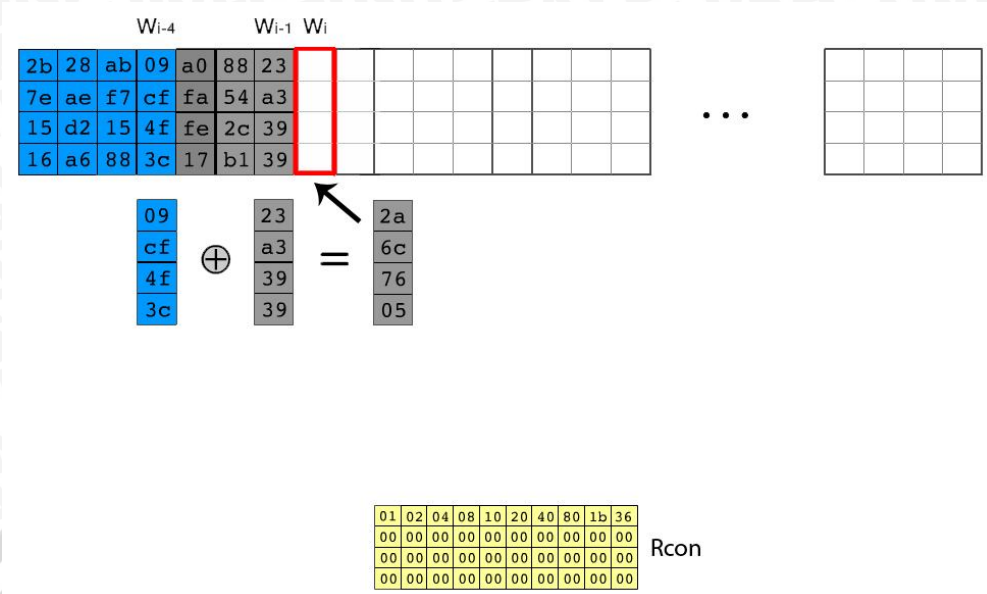

Gambar 2.13 RootWord dan SubBytes dengan S-box



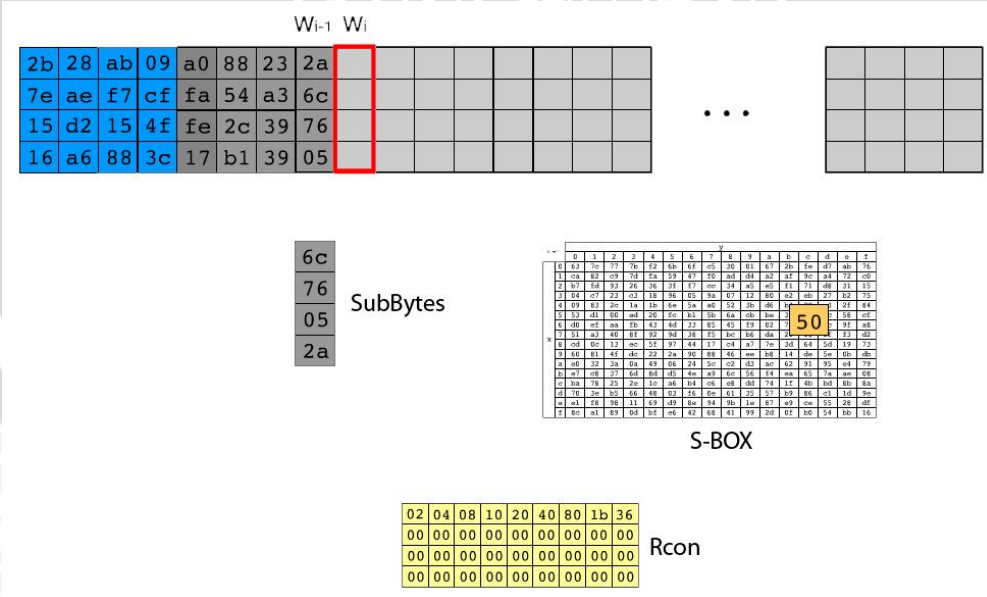

Gambar 2.14 Penambahan dengan $W_{i-4}$ dan *Rcon*



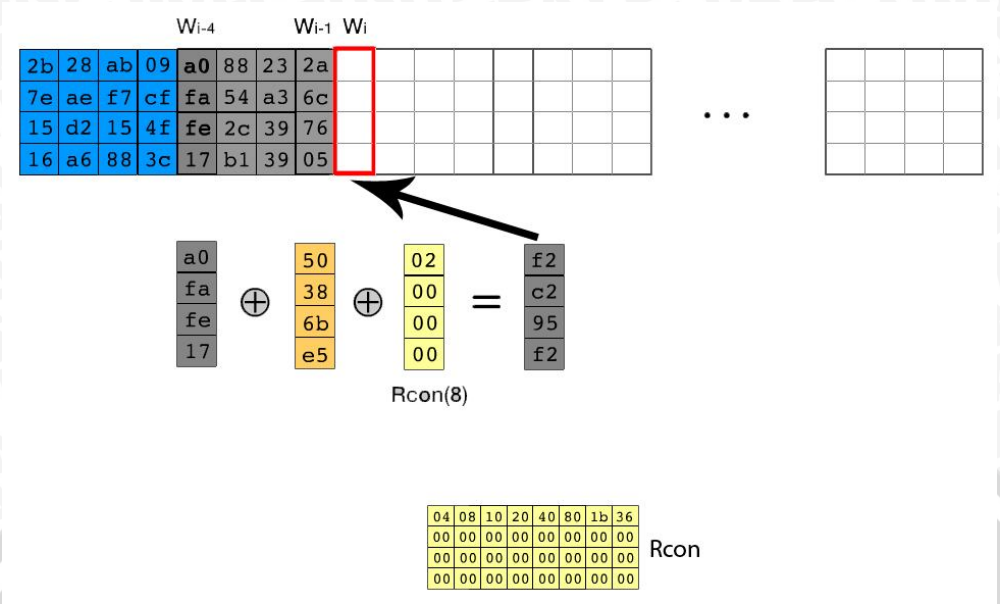

Gambar 2.15 Penambahan dengan Wi-4




Gambar 2.16 Penambahan dengan $W_{i-4}$




Gambar 2.17 RootWord dan SubBytes dengan S-box



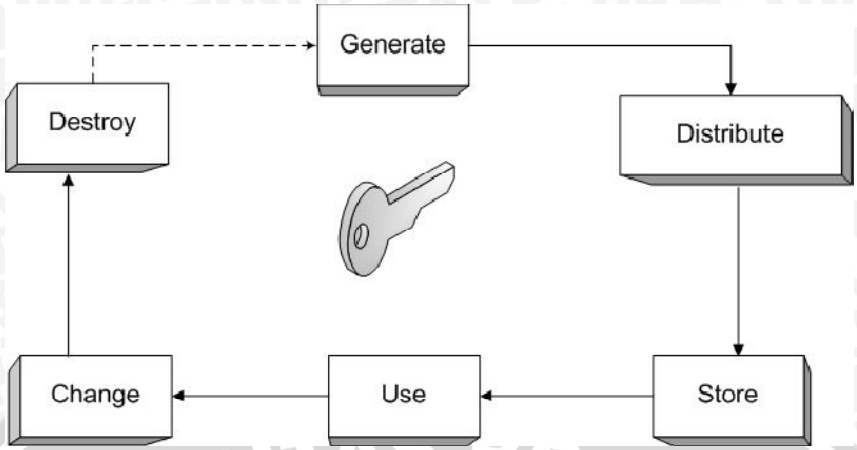

*Gambar* 2.18 Penambahan dengan $W_{i-4}$ dan *Rcon*

Langkah diatas dilakukan berulang-ulang hingga didapatkan hasil kunci seperti gambar dibawah ini :

*Gambar* 2.19 Hasil kunci putaran pada AES

## 2.6. Manajemen Kunci

Kunci Kriptografi merupakan elemen penting yang perlu dikelola secara baik. Dalam mengelola kunci Kriptografi, perlu mengakomodir alur yang dilalui sebuah kunci dari mulai diciptakan, digunakan dan kemudian (harus) dihancurkan. Alur ini disebut Daur Hidup Pengelolaan Kunci (*Key Management Life Cycle*).




*Gambar* 2.20 Daur Hidup Kunci

Pada umumnya setiap kunci pada akhirnya akan diganti dengan kunci lain. Jadi, keseluruhan fase membentuk siklus (lingkaran) karena penghancuran kunci biasanya diikuti dengan penggantiannya dengan kunci baru (garis putus-putus). Selain dikelola, Kunci perlu dilindungi selama fase daur hidupnya. [MNR-04] menyatakan, tujuan pengelolaan kunci adalah untuk menjaga keamanan dan integritas kunci pada semua fase di dalam daur hidupnya.

### 2.6.1. Pembangkitan Kunci

Pembangkitan Kunci (*key generation*) pada algoritma simetri jauh lebih mudah daripada pembangkitan kunci pada algoritma asimetri. Karena kunci algoritma simetri umumnya berupa rangkaian bit atau rangkaian karakter. Masalahnya adalah bagaimana membuat kunci yang memenuhi kriteria ’acak’agar sulit ditentukan oleh algoritma pembangkit kunci. Sedangkan pada algoritma asimetri, pembangkitan kunci merupakan masalah tersendiri, karena pembangkitan kunci membutuhkan perhitungan matematis yang rumit. Selain itu, pembangkitan bilangan prima yang besar juga dibutuhkan untuk membentuk kunci. Oleh karena itu, pada algoritma asimetri dibutuhkan program khusus untuk membangkitkan kunci, dan membutuhkan sumber daya komputasi yang relatif lebih besar. Bila program hanya dapat membangkitkan bilangan prima yang terbatas, maka pihak lawan dapat membangkitkan sendiri bilangan-bilangan prima

yang terbatas itu dan menggunakannya sebagai faktor dari salah satu parameter untuk memecahkan algoritmanya.

### 2.6.2. Penyebaran Kunci

Suatu sistem pengelolaan kunci Kriptografi harus mengakomodir penyebaran kunci (*key distribution*). Penyebaran kunci dibutuhkan agar fungsi Kriptografi dapat dijalankan. Metode pertukaran kunci dengan menggunakan algoritma asimetrik (*public key*) lazim digunakan untuk mengamankan penyebaran kunci.

### 2.6.3. Penyimpanan Kunci

Penyimpanan Kunci (*key storage*) dilakukan di tempat yang aman dan tidak memungkinkan pihak lawan mengaksesnya. Oleh karena itu, penyimpanan kunci mungkin memerlukan perlindungan secara fisik (misalnya disimpan di dalam lemari besi). Alternatif lain, kunci dapat disimpan di dalam *smart card* yang hanya dapat dibaca dengan menggunakan kode rahasia (semacam nomor PIN). Kunci sebaiknya disimpan tidak dalam bentuk jelas. Ada dua solusi alternatif untuk masalah ini :

a) Kunci disimpan dengan mengenkripsinya dengan menggunakan kunci lain. Konsep ini mengarah pada *key hierarchy*, dalam hal ini setiap kunci di dalam hirarkhi digunakan untuk melindungi kunci di bawahnya.
b) Kunci dipecah menjadi beberapa komponen, setiap komponen disimpan di tempat terpisah. Jika kunci akan digunakan, maka setiap komponen direkonstruksi kembali.

### 2.6.4. Penggunaan Kunci

Penggunaan Kunci (*key usage*) disesuaikan dengan tujuannya, misalnya ada kunci yang digunakan untuk mengenkripsi pesan, dan ada kunci yang digunakan untuk mengenkripsi kunci lainnya. Supaya setiap kunci mempunyai penggunaan yang unik, maka perlu memberi label pada setiap kunci, yang dalam hal ini label menspesifikasikan penggunaan kunci. Label tersebut menspesifikasikan ‘kunci untuk mengenkripsi data’, ‘kunci untuk mengenkripsi kunci’, ‘kunci untuk pembangkitan bilangan acak’, dan sebagainya. Untuk

algoritma asimetrik, pengguna perlu memberi label untuk sepasang kunci yang terdiri dari *public key* dan *private key*.

### 2.6.5. Penggunaan Kunci

Kunci sebaiknya diubah (*key change*) secara periodik dan teratur. Sistem pengelolaan kunci yang baik harus mempunyai kemampuan untuk mengubah kunci. Kunci diubah secara teratur untuk membatasi lama keberadaanya dan mengurangi nilainya dimata penyerang.

### 2.6.6. Penghancuran Kunci

Penghancuran Kunci (*key destruction*) dilakukan apabila kunci telah diganti oleh yang baru, atau apabila dipercaya bahwa kunci telah diketahui pihak lain (*compromised*).

Kunci yang sudah tidak dibutuhkan lagi seperti ini, harus dihancurkan dengan cara yang menyeluruh agar tidak dapat dilacak kembali. Jika kunci dicatat pada media kertas, maka cara penghancurannya misalnya menggunakan alat pemotong kertas, membakarnya, atau menguburnya. Jika kunci disimpan di dalam media elektronik (seperti *compact disc*), maka cara penghancurannya dengan menghancurkannya secara fisik. Kunci yang yang disimpan pada material lain dihancurkan sedemikian rupa sehingga ia tidak mungkin ditemukan kembali secara fisik maupun secara elektronik.

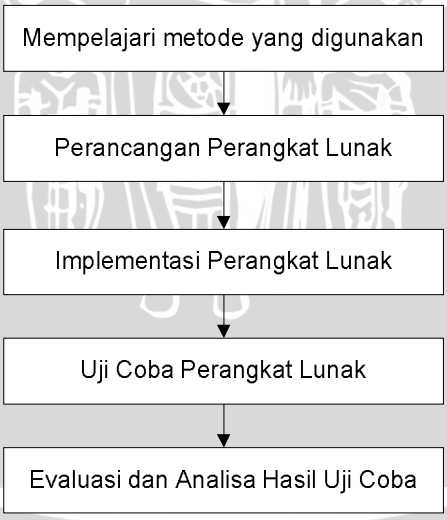
# BAB III
# METODE PENELITIAN DAN PERANCANGAN

## 3.1. Metode Penelitian

Pada bab metodologi dan perancangan ini akan dibahas mengenai langkah-langkah dan rancangan pembuatan aplikasi manajemen kunci dalam enkripsi SMS pada ponsel berbasis android. Tahapan-tahapannya adalah sebagai berikut:

1. Mempelajari literatur mengenai konsep dasar kriptografi, manajemen kunci, algoritma TDES dan algoritma AES dalam melakukan enkripsi maupun dekripsi.
2. Menganalisa dan merancang aplikasi dengan metode yang akan digunakan.
3. Implementasi perangkat lunak berdasarkan perancangan yang dilakukan
4. Uji coba enkripsi dan dekripsi SMS dengan menggunakan perangkat lunak hasil implementasi.
5. Evaluasi dan analisa berdasarkan hasil uji coba perangkat lunak.

Berikut ini merupakan blok diagram pembuatan aplikasi enkripsi SMS:



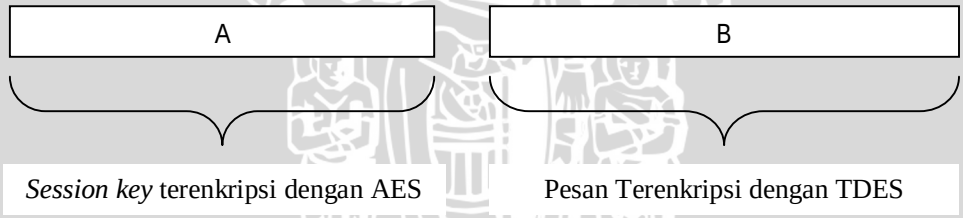

Gambar 3.1 Tahapan-Tahapan Pembuatan Perangkat Lunak

## 3.2. Perancangan Sistem

Hasil implementasi manajemen kunci pada perangkat lunak ini nantinya dapat digunakan untuk mengamankan pesan. Pengamanan pesan dilakukan dengan proses enkripsi TDES. Selain itu, kunci yang digunakan untuk proses enkripsi ini diamankan dengan enkripsi AES. Selain untuk mengamankan pesan, perangkat lunak ini juga digunakan untuk dekripsi pesan yang terenkripsi.

Proses-proses untuk melakukan enkripsi adalah sebagai berikut:

1. Pembangkitan *session key*, yang terdiri dari delapan angka acak sebagai K1, delapan angka terakhir nomor pengirim sebagai K2 dan delapan angka terakhir nomor penerima sebagai K3.
2. *Session key* kemudian dienkripsi dengan kunci K2 dan K3 menggunakan kriptografi AES kunci 128 bit.
3. Pengirim dan penerima pesan melakukan pertukaran *ciphertext session key* melalui layanan SMS atau bertukar kunci melalui *barcode scanner*.
4. Pengirim menulis pesan kemudian pesan dienkripsi menggunakan kriptografi TDES kunci 192 bit dengan menggunakan *session key*.
5. Pesan diterima dan didekripsi dengan *session key*.




Gambar 3.2 Struktur *Ciphertext* Pertukaran Kunci dan *Ciphertext* Pesan

### 3.2.1. Batasan Perangkat Lunak

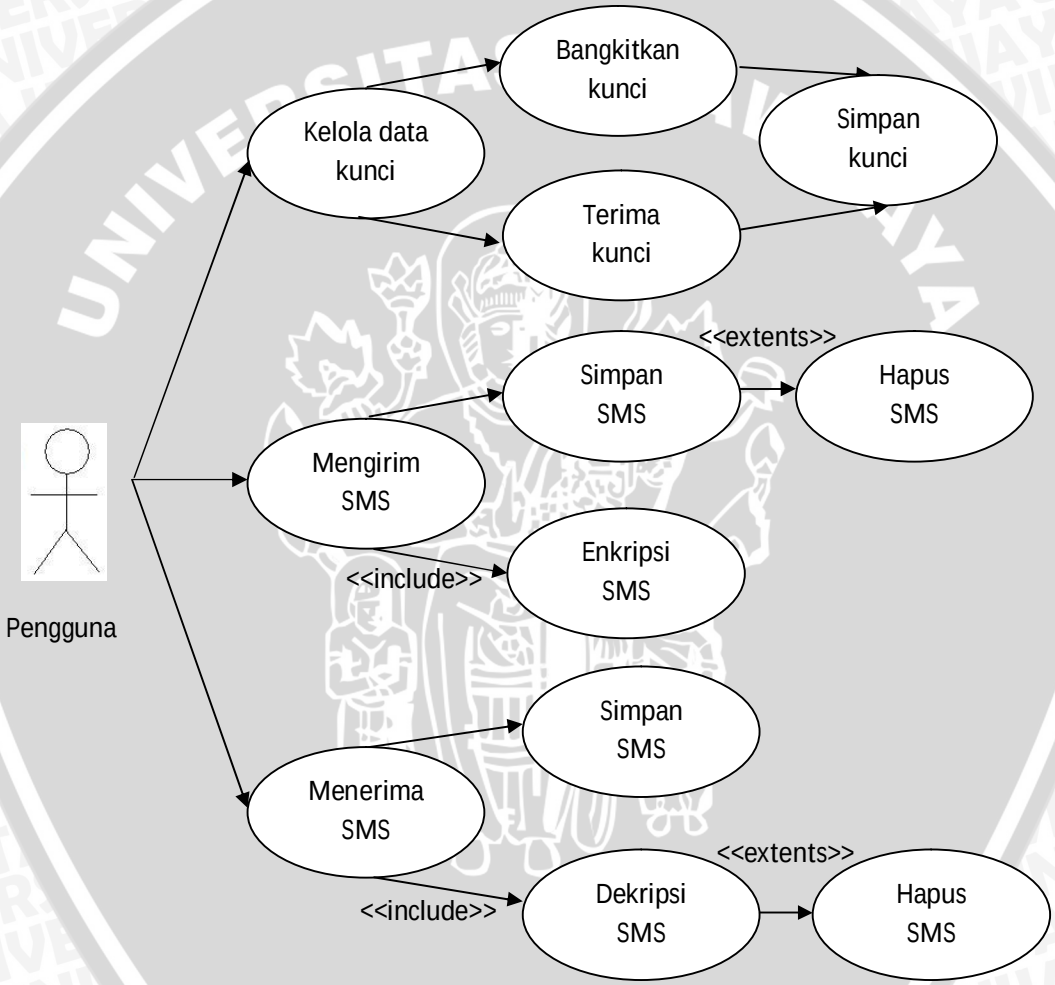
Pada perangkat lunak ini digunakan data berupa karakter ASCII. Sedangkan kunci yang digunakan dalam perangkat lunak terbagi menjadi dua dan merupakan karakter ASCII. Kunci pertama digunakan untuk proses enkripsi TDES yang terdiri dari delapan angka acak (K1), delapan angka terakhir nomor pengirim (K2) dan delapan angka terakhir nomor penerima pesan (K3). Kunci yang kedua adalah kunci privat yang digunakan untuk proses enkripsi AES dengan aturan maksimal 16 karakter ASCII dan terdiri dari K2 dan K3.

### 3.2.2. Perancangan Perangkat Lunak

Adapun proses utama dalam perangkat lunak ini adalah sebagai berikut :

#### 3.2.2.1.Diagram *Use Case*

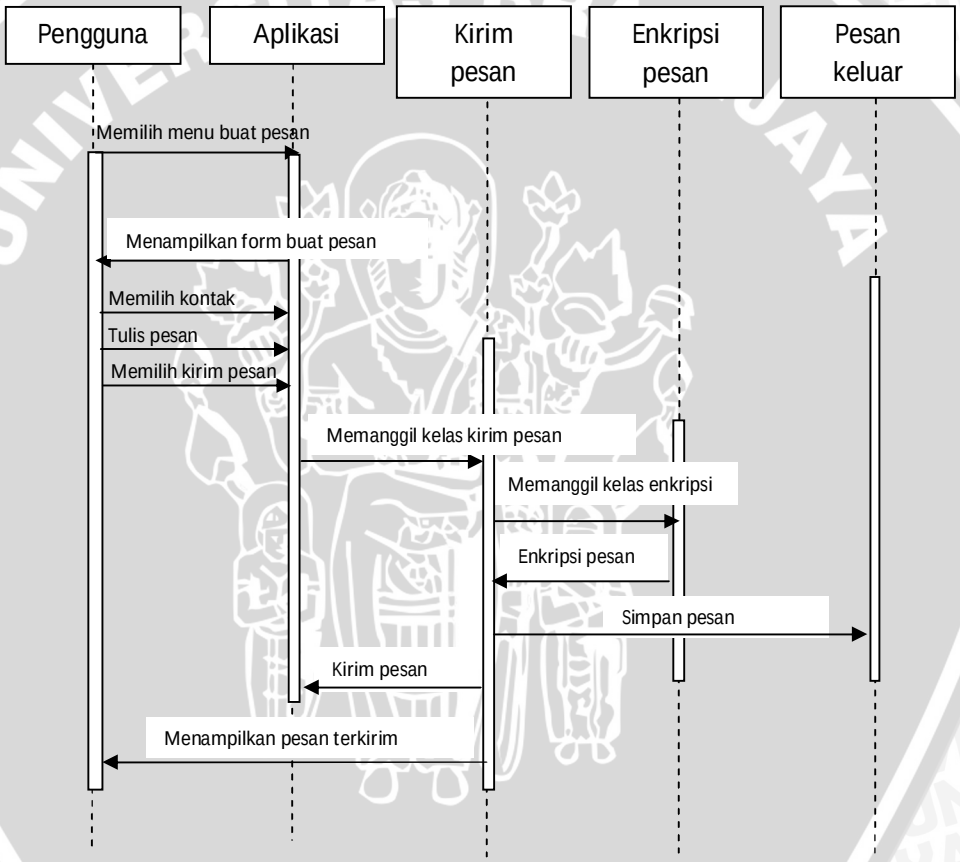
Diagram *use case* menunjukkan interaksi antara penggunan dengan entitas ekternal lainnya. *Use case* menggambarkan fungsi dari suatu sistem. Gambar 3.3 dibawah ini menunjukkan *use case* dari perangkat lunak.



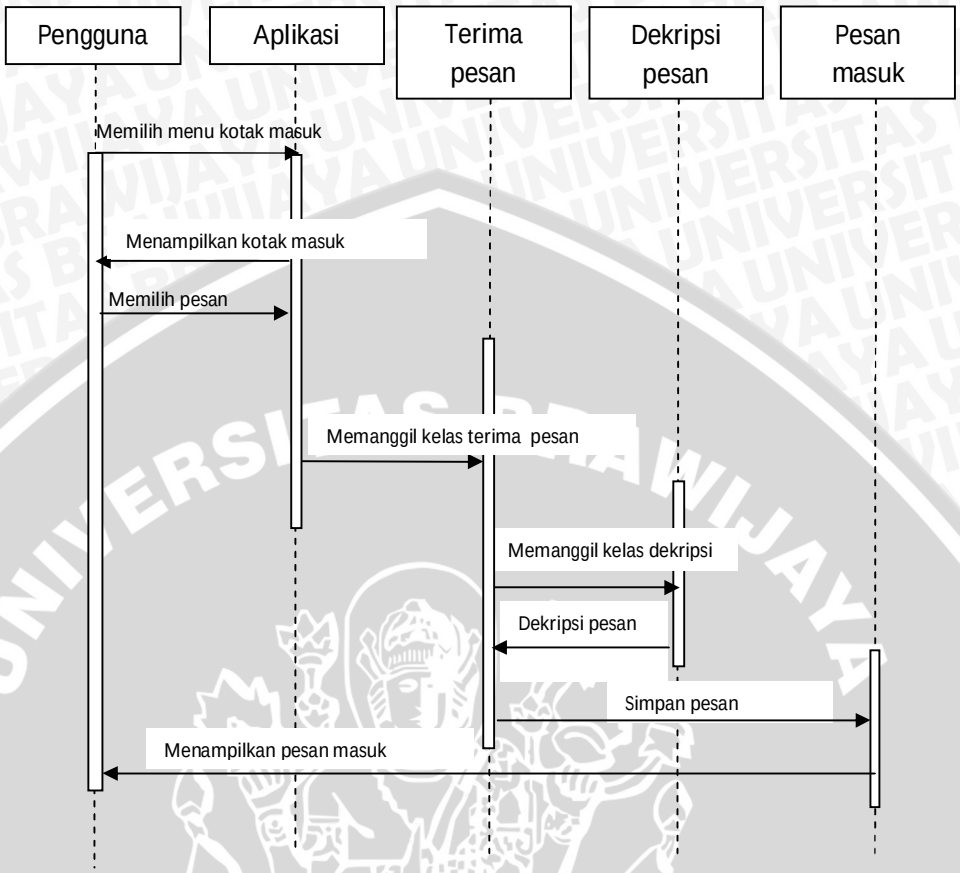

Gambar 3.3 Gambar diagram *use case*

### 3.2.2.2.*Sequence Diagram*

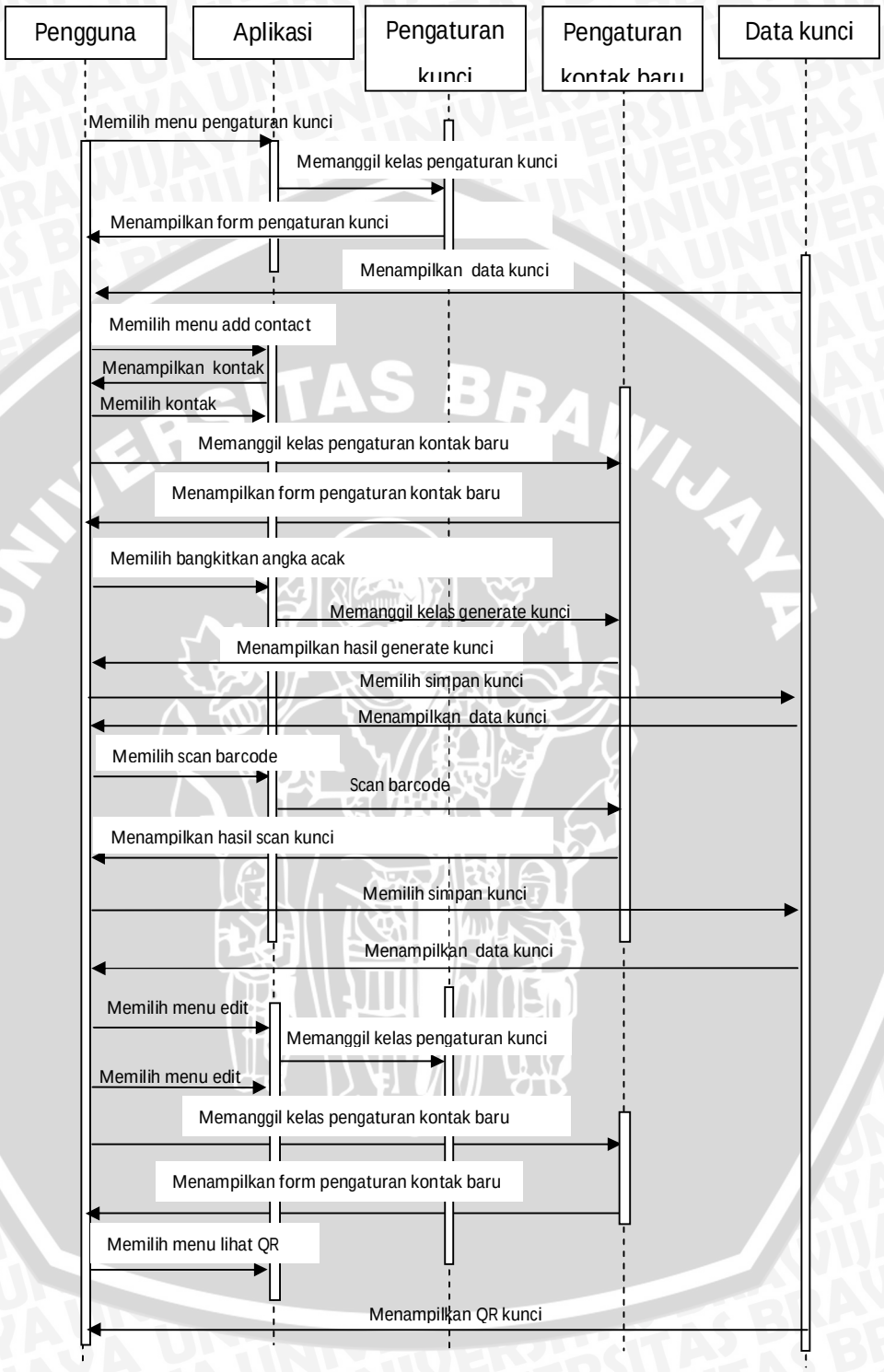
Sequence diagram merepresentasikan interaksi antar objek di dalam dan di sekitar sistem, termasuk pengguna dan antarmuka pengguna. Sequence diagram terdiri atas dimensi vertikal yang merepresentasikan waktu, dan dimensi horizontal yang merepresentasikan objek-objek terkait. Aspek penting dari sequence diagram adalah urutan waktu, yang mengindikasikan bahwa interaksi direpresentasikan secara bertahap. Sequence diagram di sini dibangun berdasarkan diagram use case pada gambar 3.3.




Gambar 3.4 Gambar *sequence diagram* kirim pesan.




Gambar 3.5 Gambar *sequence diagram* terima pesan




Gambar 3.6 Gambar *sequence diagram* pengaturan kunci

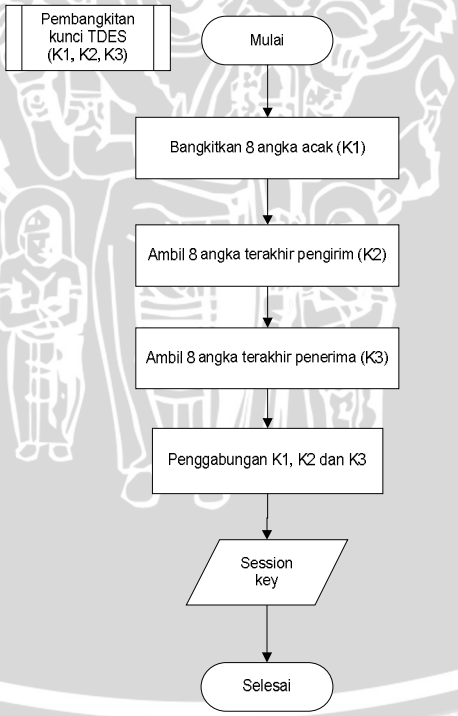
### 3.2.3. Perancangan Algoritma Enkripsi dan Dekripsi

#### 3.2.3.1 Pembentukan Kunci TDES 192 Bit

Fungsi utama kunci adalah untuk menyandikan pesan dan mengembalikan pesan yang dienkripsi menjadi pesan asli. Selain itu kunci juga dapat mengamankan pesan, jadi jika kunci tidak cocok maka pesan terenkripsi tidak dapat dikembalikan seperti semula menjadi pesan asli. Langkah-langkah dalam pembentukan kunci untuk enkripsi pesan adalah sebagai berikut :

1. Sistem membangkitkan delapan angka acak sebagai K1.
2. Sistem mengambil delapan angka terakhir dari nomor pengirim sebagai K2.
3. Sistem mengambil delapan angka terakhir dari nomor penerima sebagai K3.
4. *Session key* dibentuk dengan menggabungkan K1, K2 dan K3.

Berikut *flowchart* untuk pembentukan kunci yang ditunjukkan pada Gambar 3.7.



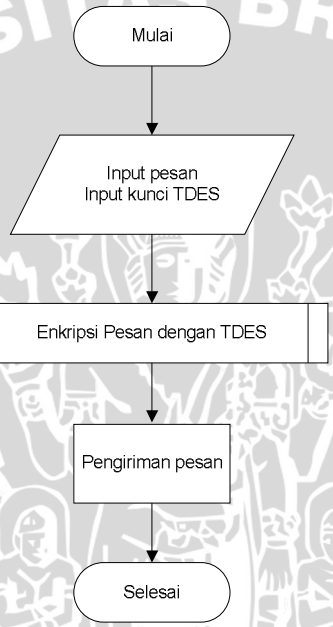

Gambar 3.7 *Flowchart* Pembentukan Kunci

### 3.2.3.2 Enkripsi dengan TDES 192 Bit

Langkah-langkah yang dilakukan pada enkripsi TDES ini adalah :

1. Pengguna memasukkan pesan yang akan dienkripsi.
2. Sistem mengambil *session key*.
3. Sistem mengenkripsi pesan dengan TDES dengan *session key* 192 bit.
4. *Ciphertext* pesan dihasilkan dan siap dikirim.

Berikut *flowchart* untuk enkripsi dengan TDES yang ditunjukkan pada Gambar 3.8.



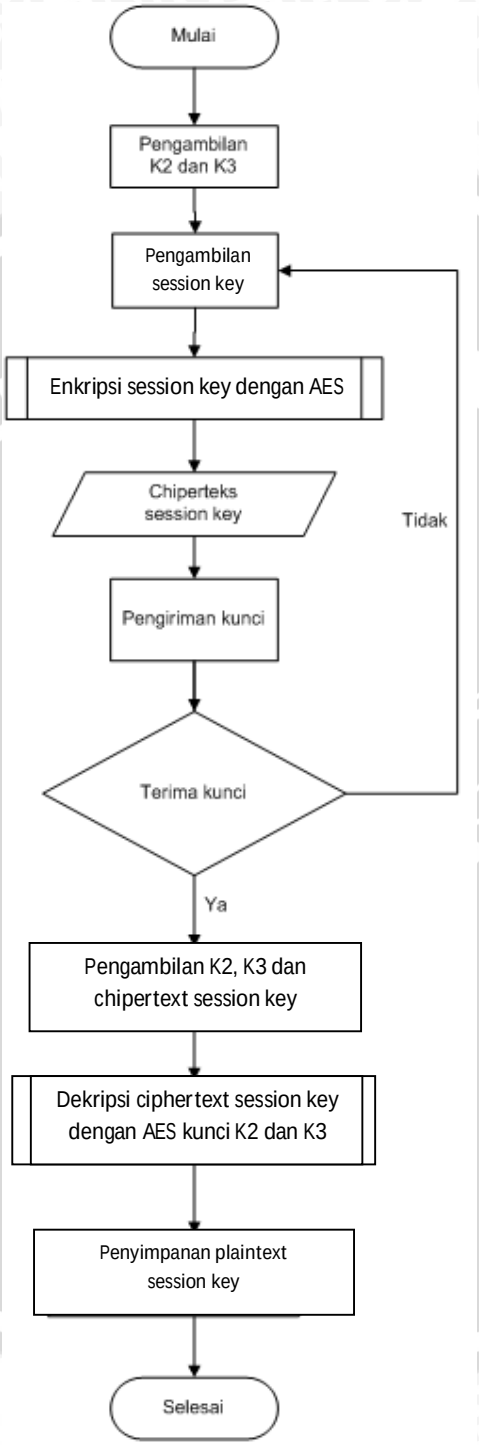

Gambar 3.8 *Flowchart* Enkripsi Pesan

### 3.2.3.3 Pengiriman Kunci TDES melalui SMS

Langkah-langkah yang dilakukan pada pertukaran kunci adalah :

1. Sistem mengambil K2 dan K3.
2. Sistem mengambil *session key* dari *database* kunci.
3. Sistem mengenkripsi *session key* dengan AES kunci K2 dan K3.
4. *Ciphertext session key* dihasilkan kemudian dikirim melalui SMS.
5. Apabila penerima pesan menerima kunci, maka sistem akan mengambil *ciphertext session key* dari pesan masuk.
6. Sistem akan mendekripsi *ciphertext session key* kemudian hasilnya disimpan dalam *database* kunci.

Berikut *flowchart* untuk pengiriman kunci melalui sms yang ditunjukkan pada Gambar 3.9.



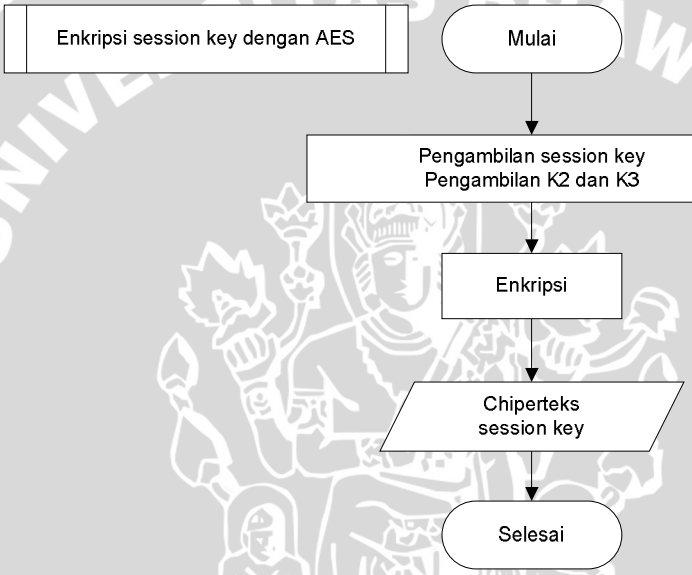

Gambar 3.9 *Flowchart* pengiriman *session key* melalui SMS

### 3.2.3.4 Enkripsi Session key dengan AES 128 Bit

Langkah-langkah yang dilakukan pada enkripsi K1 dan K2 adalah :

1. Sistem mengambil K2 dan K3.
2. Sistem mengenkripsi *session key* dengan AES dengan kunci privat 128 bit K2 dan K3
3. *Ciphertext session key* dihasilkan dan siap dikirim.

Berikut *flowchart* untuk enkripsi *session key* yang ditunjukkan pada Gambar 3.10.



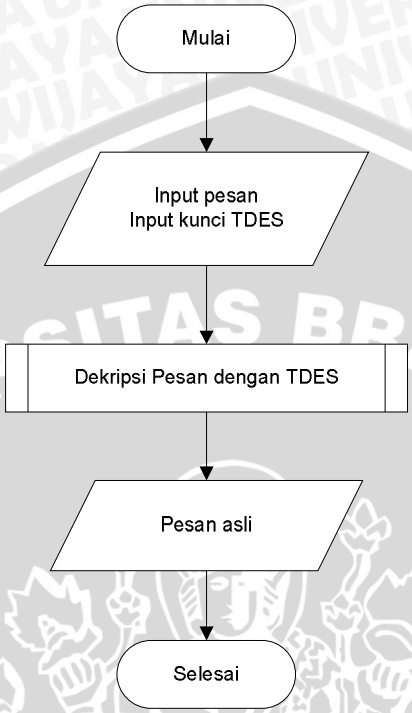

Gambar 3.10 Flowchart Enkripsi session key.

### 3.2.3.5 Dekripsi dengan TDES 192 Bit

Langkah-langkah yang dilakukan pada dekripsi *ciphertext* pesan adalah :

1. Sistem mengambil pesan yang akan didekripsi.
2. Sistem mengambil *session key*.
3. Sistem mendekripsi pesan dengan *session key* 192 bit.
4. *Plaintext* pesan dihasilkan.

Berikut *flowchart* untuk dekripsi dengan TDES yang ditunjukkan pada Gambar 3.11.



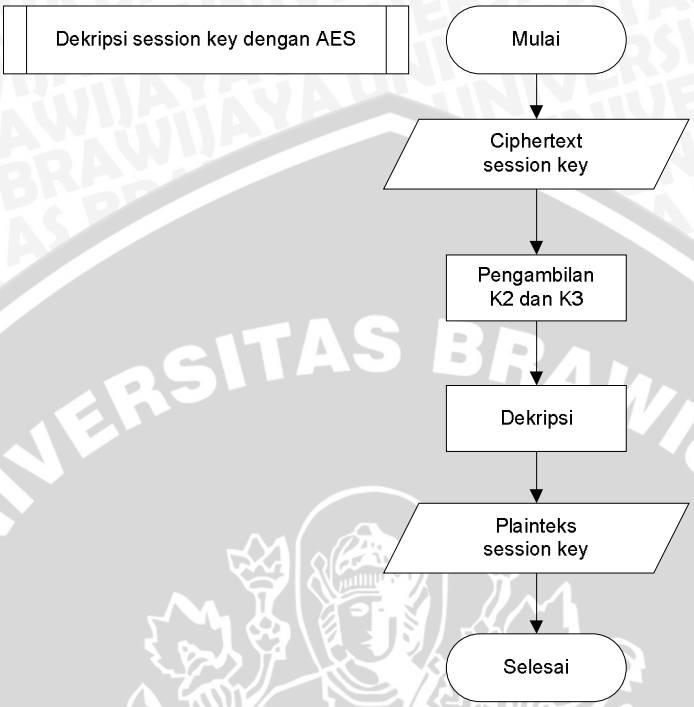

Gambar 3.11 *Flowchart* Dekripsi Pesan

### 3.2.3.6 Dekripsi dengan AES 128 bit

Langkah-langkah yang dilakukan pada dekripsi *ciphertext* K1 dan K2 ini adalah :

1. Sistem menerima *ciphertext session key*.
2. Sistem mendekripsi *sesssion key* dengan kunci privat 128 bit K2 dan K3
3. *Plaintext session key* dihasilkan dan siap disimpan.

Berikut *flowchart* untuk dekripsi dengan AES yang ditunjukkan pada Gambar 3.12.



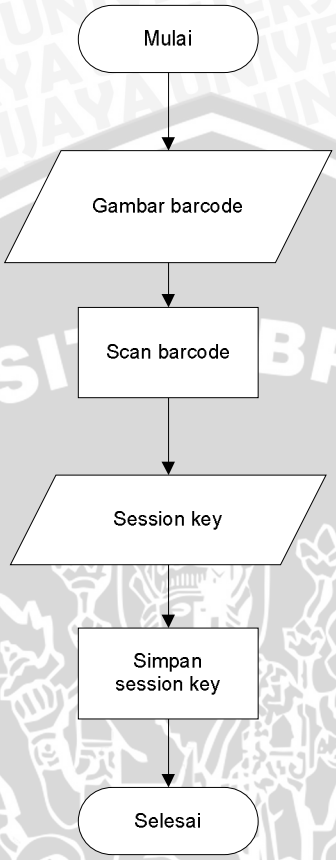

Gambar 3.12 *Flowchart* dekripsi *session key*

### 3.2.3.7 Pertukaran Kunci dengan *Barcode Scanner*

Langkah-langkah yang dilakukan pada pertukaran kunci adalah :

1. Arahkan kamera ke *barcode* kunci.
2. Sistem membaca *barcode* kunci.
3. *Session key* dihasilkan kemudian disimpan.

Berikut *flowchart* untuk pertukaran kunci dengan *barcode scanner* yang ditunjukkan Gambar 3.13.




Gambar 3.13 *Flowchart* dekripsi *session key*

### 3.2.4. Perancangan Antar Muka

Antar muka perangkat lunak ini terdiri dari beberapa bagian yaitu :

#### 3.2.4.1 Antar Muka Menu Utama

Berikut merupakan rancangan antar muka perangkat lunak menu utama yang ditunjukan pada Gambar 3.14:

Gambar 3.14 Antar Muka Menu Utama

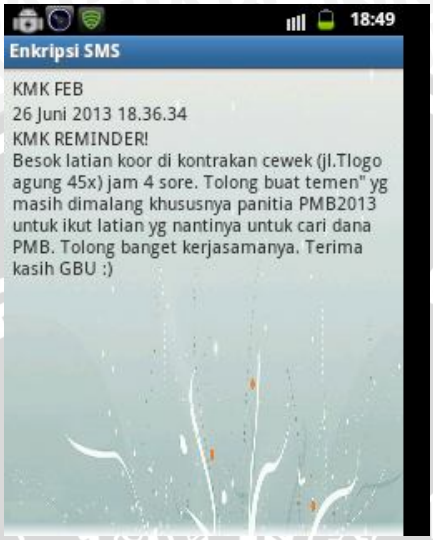
### 3.2.4.2 Antar Muka Kirim Pesan

Berikut merupakan rancangan antar muka perangkat lunak pada menu kirim pesan yang ditunjukan pada Gambar 3.15:

Gambar 3.15 Antar Muka Tulis Pesan

### 3.2.4.3 Antar Muka Kotak Masuk

Berikut merupakan rancangan antar muka perangkat lunak pada menu kotak masuk yang ditunjukan pada Gambar 3.16:




Gambar 3.16 Antar Muka Kotak Masuk

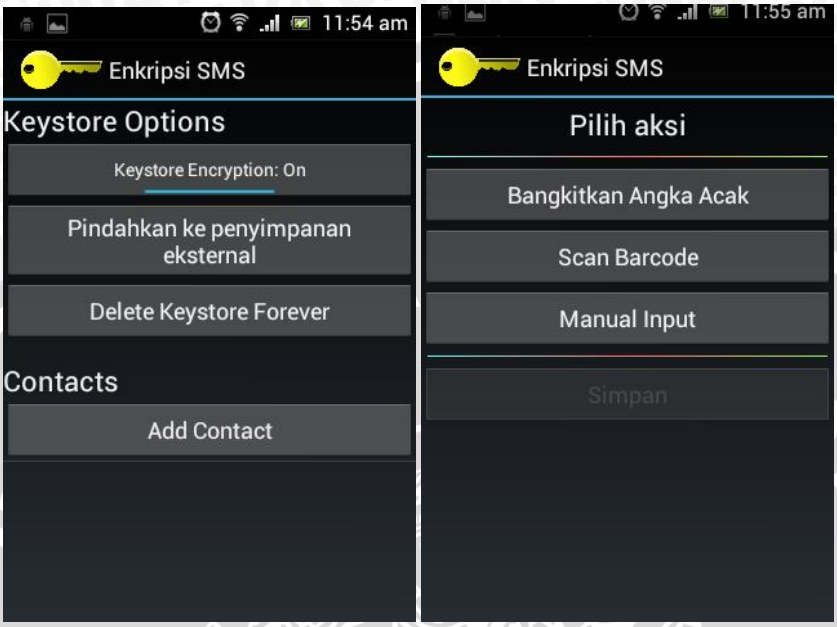
### 3.2.4.4 Antar Muka Kotak Keluar

Berikut merupakan rancangan antar muka perangkat lunak pada menu kotak keluar yang ditunjukan pada Gambar 3.17:

Gambar 3.17 Antar Muka Pesan Kotak Keluar

### 3.2.4.5 Antar Muka Pengaturan Kunci

Berikut merupakan rancangan antar muka perangkat lunak pada menu pengaturan kunci yang ditunjukan pada Gambar 3.18:




Gambar 3.18 Antar Muka Pengaturan Kunci

# BAB IV
# IMPLEMENTASI

Pada bab IV ini akan dijelaskan mengenai implementasi dari seluruh proses yang sudah dirancang pada bab sebelumnya.

## 4.1. Lingkungan Implementasi

Implementasi merupakan proses transformasi representasi rancangan ke dalam bahasa pemrograman yang dapat dimengerti oleh ponsel. Pada bab ini, lingkungan implementasi yang akan dijelaskan meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

### 4.1.1. Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan dan pengujian sistem manajemen kunci pesan SMS ini adalah :

1. Sony Ericsson Xperia Mini ST15i dengan spesifikasi CPU 1 GHz, RAM 512MB, Android OS, v4.0 (*Ice Cream Sandwich*), layar 3" dan kamera 5Mpx.
2. Sony Ericsson Xperia Ray ST18i dengan spesifikasi CPU 1 GHz, RAM 512MB, Android OS, v4.0 (*Ice Cream Sandwich*), layar 3,3" dan kamera 8Mpx.
3. Samsung Galaxy Fit S5670 dengan spesifikasi CPU 600 MHz, RAM 280MB, Android OS, v2.3 (*Gingerbread*), layar 3,3" dan kamera 5Mpx.
4. Notebook.

Perangkat keras terebut akan difungsikan sebagai tempat pembuatan perangkat lunak dan untuk menjalankan penelitian.

### 4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem kriptografi dan manajemen kunci ini adalah :

1. Sistem operasi *Windows 7™ Ultimate 32-bit* sebagai tempat pembuatan perangkat lunak.
2. *Eclipse Android Developer Tools v21.1.0-569685* dalam pembuatan sistem kriptografi dan manajemen kunci pesan SMS.
3. *Whatsapp* dalam proses pembangkitan kunci.
4. *Barcode Scanner* dalam proses pengiriman dan penerima kunci.

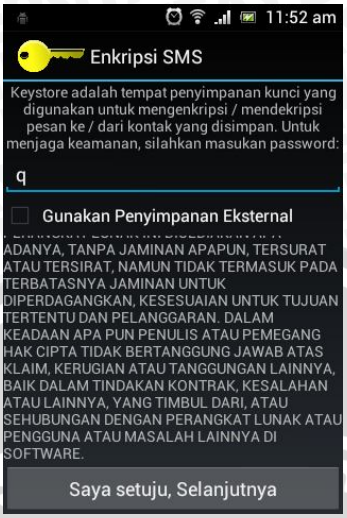
## 4.2 Implementasi Perangkat Lunak

Berdasarkan rancangan perangkat lunak yang dikemukakan pada Bab III maka dihasilkan fungsi-fungsi utama perangkat lunak sebagai berikut :

### 4.2.1. Instalasi Perangkat Lunak

Untuk dapat menggunakan perangkat lunak ini pertama-tama pasang perangkat lunak ini pada ponsel android. Pada proses instalasi ini, perangkat menjalankan kelas SetupActivity.java serta dan dihasilkan antar muka sebagai seperti pada gambar 4.1. Adapun inti dari proses SetupActivity.java adalah sebagai berikut :

```
String text = password.getText().toString();

if(!text.equals("")) {
    MessageDigest md;
    md = MessageDigest.getInstance("SHA-256");
    md.update(text.getBytes("UTF-8"));
    KeyManager.getInstance().setKeyStoreKey(md.digest());
    KeyManager.getInstance().setPasswordProtected(true);
} else {
    KeyManager.getInstance().setPasswordProtected(false);
}

KeyManager.getInstance().setInternalStorage(!external.isChecked());
```

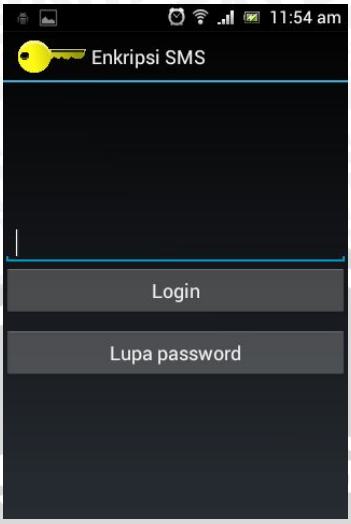



Gambar 4.1 Antar Muka Proses Instalasi

### 4.2.2. Menu *Log In*

Setelah proses instalasi selesai, untuk dapat menggunakan perangkat lunak ini dilakukan proses *log in*. Pada proses *log in* perangkat lunak memanggil kelas SetupActivity.java dan dihasilkan antar muka pada gambar 4.2. Inti dari proses ini sebagai berikut :

```
try {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(loginText.getText().toString().getBytes("UTF-8"));
    KeyManager.getInstance().setKeyStoreKey(md.digest());
    KeyManager.getInstance().load(SetupActivity.this);

    finish();

} catch(Exception e) {
    Toast.makeText(SetupActivity.this, R.string.login_failed, Toast.LENGTH_SHORT).show();
}
```




Gambar 4.2 Antar Muka Proses *Log in*

### 4.2.3. Antar Muka Menu Utama

Setelah masuk ke dalam perangkat lunak, maka akan terdapat beberapa menu yang akan diproses kelas MainMenuActivity.java seperti yang ditampilkan antar muka menu utama dibawah ini. Inti dari proses MainMenuActivity.java ini adalah :

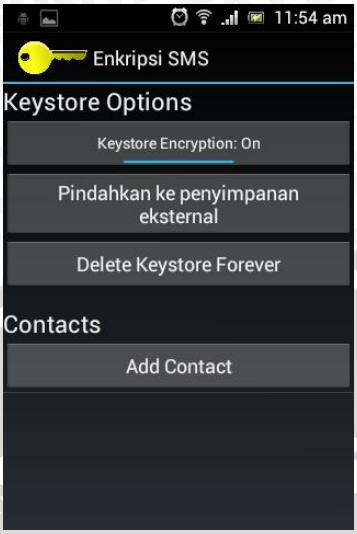
```
((Button) findViewById(R.id.sms))
        .setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                MainMenuActivity.this.startActivity(new Intent(
                        MainMenuActivity.this, SMSActivity.class))
            }
        });

((Button) findViewById(R.id.keyManage))
        .setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                MainMenuActivity.this.startActivity(new Intent(
                        MainMenuActivity.this,
                        KeyManagementActivity.class));
            }
        });

((Button) findViewById(R.id.datapesan))
        .setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                MainMenuActivity.this.startActivity(new Intent(
                        MainMenuActivity.this, Inbox.class));
            }
        });
```

```
((Button) findViewById(R.id.sent))
        .setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                MainMenuActivity.this.startActivity(new Intent(
                        MainMenuActivity.this, Sent_item.class));
            }
        });

((Button) findViewById(R.id.exit))
        .setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                MainMenuActivity.this.finish();
            }
        });
```

Gambar 4.3 Antar Muka Menu Utama

### 4.2.4. Pengaturan Kunci

Untuk dapat menggunakan semua layanan dalam perangkat lunak ini, atur kunci terlebih dahulu dengan memilih menu pengaturan pada menu utama. Proses ini ditangani oleh kelas KeyManagementActivity yang terdiri dari beberapa proses yang terdapat pada antar muka sebagai berikut :




Gambar 4.4 Antar Muka Pengaturan Kunci

Tombol pindahkan ke penyimpanan eksternal berfungsi memindah semua data aplikasi ke memori penyimpanan ekternal.

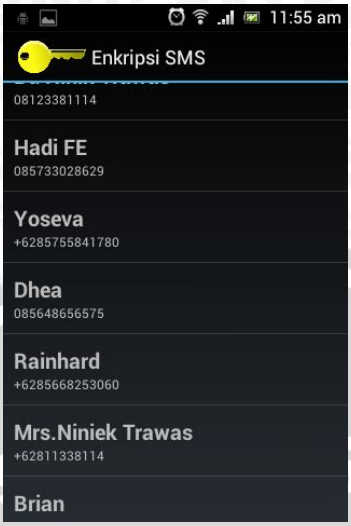
```
public void onClick(View v) {
    if(KeyManager.getInstance().swap(KeyManagementActivity.this)) {
        moveKeystore.setText((KeyManager.getInstance().isInternalStorage())
                ? R.string.move_to_external : R.string.move_to_internal);
    } else {
        Toast.makeText(KeyManagementActivity.this, R.string.move_error, Toast.LENGTH_SHORT).show();
    }
}
```

Tombol *Delete Keytore Forever* berfungsi untuk menghapus semua data kunci. Berikut inti dari proses penghapusan semua data kunci :

```
public void onClick(DialogInterface dialog, int which) {
    Context context = KeyManagementActivity.this;
    KeyManager.getInstance().delete(context);

    context.startActivity(new Intent(context, SetupActivity.class));
    finish();
}
```

Tombol *Add Contact* berfungsi untuk menambah kunci dari kontak yang dipilih. Pilih tombol *Add Contact* maka akan dilakukan proses pengambilan data kontak dari ponsel dan tampilan antar mukanya seperti pada gambar 4.5.

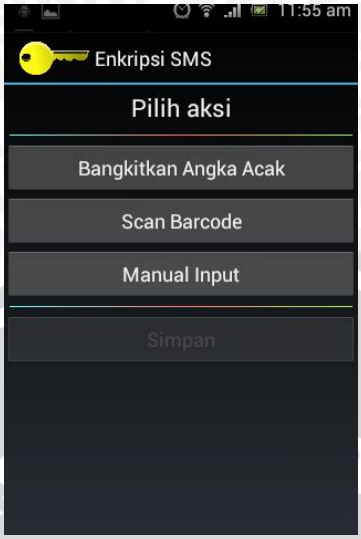
```
public void onClick(View v) {
    Intent intent = new Intent(KeyManagementActivity.this, DisplayContactsActivity.class);
    startActivityForResult(intent, SELECT_CONTACT);
}
```




Gambar 4.5 Antar Muka *Add Contact*

Setelah salah satu kontak dipilih maka akan muncul antar muka seperti pada gambar 4.6 yang akan diproses oleh kelas NewContactActivity.java yang akan melakukan proses penyimpanan kunci.

```
private void setKey(byte[] newKey) {
    //Set kunci baru
    if(newKey != null && newKey.length == AES.AES_KEY_SIZE) {
        key = newKey;
    } else {
        Toast.makeText(this, R.string.invalid_key, Toast.LENGTH_SHORT).show();
    }
    saveButton.setEnabled(key != null);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (data != null) {
        switch(requestCode) {
        case GENERATE_KEY_RESULT:
            if(resultCode == Activity.RESULT_OK){
                setKey(data.getByteArrayExtra("data"));
            }
            break;
        case SCAN_CODE:
            if(resultCode == Activity.RESULT_OK) {
                IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
                byte[] values = null;
                try {
                    values = Base64.decode(result.getContents());
                    Toast.makeText(this, values.toString() , Toast.LENGTH_SHORT).show();
                } catch(Exception e) {
                    e.printStackTrace();
                }
                setKey(values);
            }
            break;
        }
    }
}
```




Gambar 4.6 Antar Muka Pilih Aksi

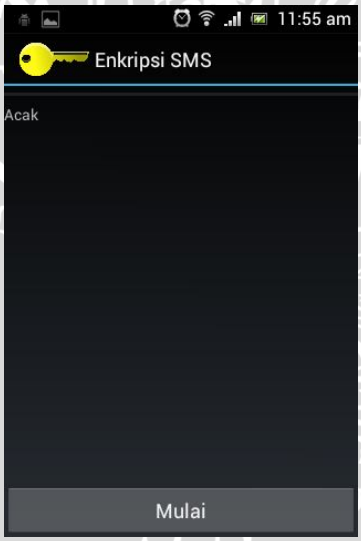
Tombol *Scan Barcode* berfungsi untuk mengambil kunci dari ponsel lain yang telah membangkitkan kunci. Adapun proses scan barcode ini adalah sebagai berikut :

```
public void onClick(View v) {
    IntentIntegrator integrator = new IntentIntegrator(NewContactActivity.this);
    integrator.initiateScan(IntentIntegrator.QR_CODE_TYPES, SCAN_CODE);
}
```

Tombol bangkitkan angka acak berfungsi untuk membangkitkan kunci yang akan digunakan pada proses kriptografi TDES. Antar muka proses ini ditampilkan pada gambar 4.7. Proses ini memanggil kelas GenerateActivity.java dan menghasilkan Generate_key_result. Hasil ini akan diproses kembali oleh kelas KeyManagementActivity.java. Berikut proses pada GenerateActivity.java.

Tombol manual input berfungsi untuk mengambil kunci dari kotak masuk. Bila kunci sudah dikirim oleh lawan komunikasi maka sistem akan mencari kunci dari kotak masuk. Kunci ini akan didekripsi dan hasilnya akan disimpan dalam *database* kunci.

```
AccountManager am = AccountManager.get(getApplicationContext());
Account[] accounts = am.getAccounts();

for (Account ac : accounts) {
 String acname = ac.name;
 String actype = ac.type;

 if(actype.equals("com.whatsapp")){
      String phoneNumber = acname;
        final Random myRandom = new Random();
        String rand = new String((String.valueOf(myRandom.nextInt(99999999))));

        StringBuilder sb = new StringBuilder();
        sb = sb.append(phoneNumber);
        String pengirim = sb.substring(phoneNumber.length()-8).toString();

        StringBuilder rb = new StringBuilder();
        rb = rb.append(nomor);
        String penerima = rb.substring(nomor.length()-8).toString();


        String kunci = (rand+pengirim+penerima);

        byte[] data1 = kunci.getBytes();

        if(kunci != null && kunci.length() == AES.AES_KEY_SIZE) {
            data = data1 ;
          Intent intent = new Intent();
          intent.putExtra("data", data);
          GenerateActivity.this.setResult(RESULT_OK, intent);
          GenerateActivity.this.finish();
          Toast.makeText(GenerateActivity.this, kunci +"##" + data.toString() , Toast.LENGTH_LONG).show();
        }
```

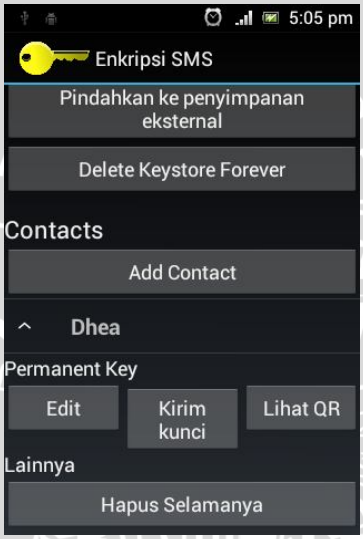



Gambar 4.7 Antar Muka Proses Membangkitkan Angka Acak

Tombol mulai berfungsi melakukan proses pembangkitan kunci. Bila kunci telah selesai dihasilkan maka kembali pada antar muka *Add Contact* dan tombol simpan pada menu *Add Contact* menjadi *enable* (tersedia) . Proses penyimpanan kuncinya adalah sebagai berikut :

```
saveButton = (Button) findViewById(R.id.saveButton);
saveButton.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        Intent resultIntent = new Intent();
        resultIntent.putExtra("address", contactAddress);
        resultIntent.putExtra("name", contactName);
        resultIntent.putExtra("key", key);
        setResult(Activity.RESULT_OK, resultIntent);
        finish();
    }
});
```

Setelah menyimpan kunci maka akan kembali pada antar muka pengaturan dengan tambahan antar muka daftar kontak yang telah memiliki kunci. Antar muka pengaturan yang baru :




Gambar 4.8 Antar Muka Pengaturan yang Telah Diperbaharui

Pada menu list kontak yang telah memiliki kunci terdapat pilihan menu edit, lihat QR dan hapus selamanya. Tombol edit akan mengarahkan pengguna pada menu pilih aksi atau proses pada kelas NewContactActivity.java. Tombol hapus selamanya berfungsi menghapus data kunci pada kontak tersebut. Prosesnya adalah sebagai berikut :

```
Button delete = (Button) view.findViewById(R.id.delete);
delete.setOnClickListener(new ClickListener(contact) {
    public void onClick(View v) {
        Dialogs.showConfirmation(context, R.string.confrimation_msg, new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                KeyManager.getInstance().getLookup().remove(contact.address);
                populate();
            }
        });
    }
});
```

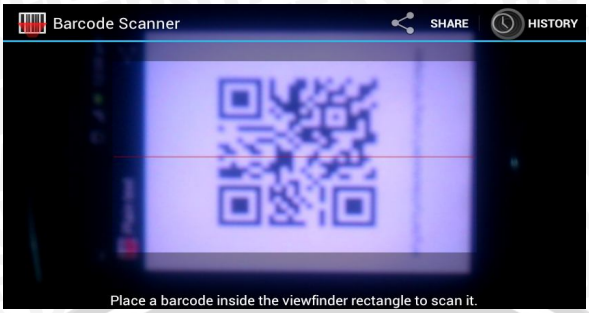
Tombol lihat QR berfungsi menampilkan barcode kunci. Proses lihat QR ini adalah :

```
Button viewButton = (Button) view.findViewById(R.id.view);
viewButton.setOnClickListener(new ClickListener(contact) {
    public void onClick(View v) {
        Key key = KeyManager.getInstance().getLookup().get(contact.address);
        //menampilkan kode QR
        (new IntentIntegrator(context)).shareText(new String(Base64.encode(key.key)));
    }
});
```

Gambar 4.9 Contoh Antar Muka Lihat QR

Gambar *barcode* kunci ini digunakan untuk pertukaran kunci antara 2 pengguna perangkat lunak yang bertindak sebagai pengirim dan penerima pesan SMS. Contoh pengambilan kunci dengan *barcode scanner* ini adalah dengan menggunakan kamera ponsel. Tampilannya adalah sebagai berikut.

Jika memilih kirim kunci maka kunci dari kontak. Kunci tersebut akan dienkripsi dan dikirim melalui SMS kepada lawan komunikasi.

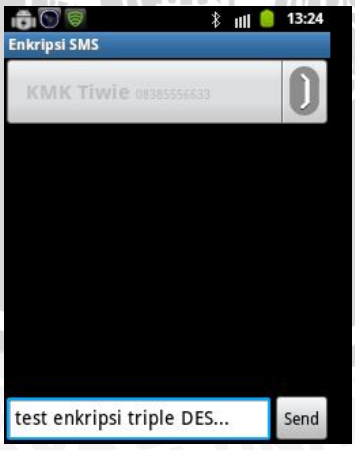



Gambar 4.10 Antar Muka Pengambilan Kunci dengan *Barcode Scanner*

### 4.2.5. Tulis Pesan

Setelah pengaturan kunci selesai maka kita dapat mengirim pesan ke pengguna yang telah terdaftar memiliki kunci kriptografi. Nantinya pesan akan dikirim setelah dilakukan enkripsi dengan menggunakan kunci yang telah terdaftar. Antar muka dari buat pesan ditunjukkan pada gambar 4.11. Proses tulis pesan ini mamanggil kelas SMSActivity.java yang prosesnya adalah sebagai berikut :

```
try {
    test = new TripleDES(key.key);
    sendSMS(currentAddress, test.encrypt(message));
    smsAdapter.populateSMS(displayName, currentAddress);
    Toast.makeText(SMSActivity.this, message ,
            Toast.LENGTH_SHORT).show();
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Toast.makeText(SMSActivity.this, R.string.error_send,
            Toast.LENGTH_SHORT).show();
}
```

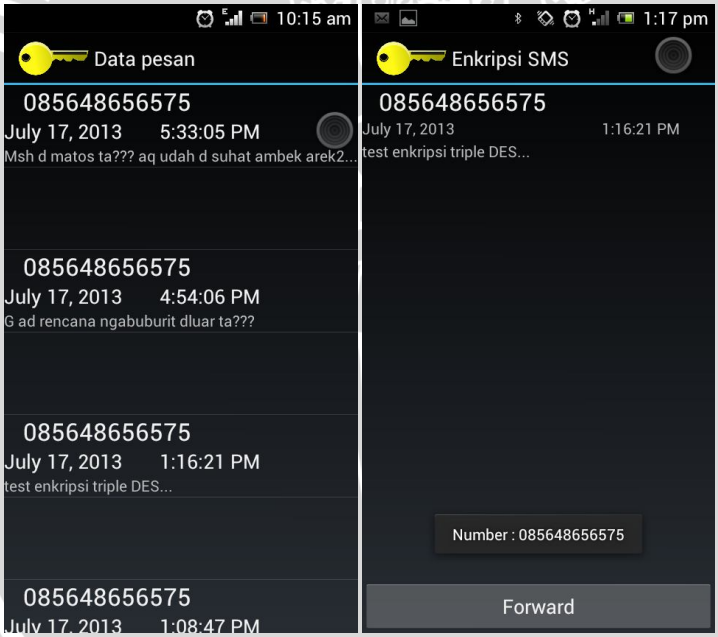



Gambar 4.11 Antar Muka Buat Pesan

### 4.2.6. Pesan Masuk

Pesan yang masuk dari daftar kontak yang telah memiliki kunci akan didekripsi sesuai kunci yang terdaftar. Antar muka daftar pesan adalah seperti gambar 4.12. Apabila salah satu pesan dipilih maka akan tampil pesan yang dipilih. Adapun inti dari proses dekripsi pesan ini adalah :

```
Key key = KeyManager.getInstance().getLookup()
        .get(currentAddress);
if (key.key != null) {


    try {
        String chiper = temp.message.substring(1);
        test = new TripleDES(key.key);
        temp.message = test.decrypt(chiper);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        //e.printStackTrace();
    }
```



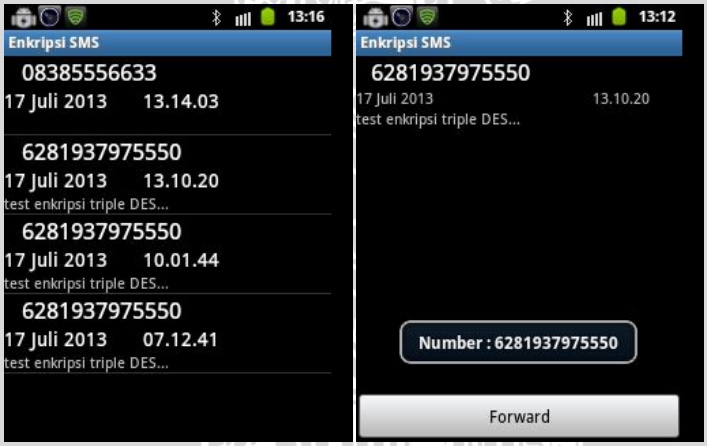

Gambar 4.12 Antar Muka Pesan Masuk

### 4.2.7. Pesan Keluar

Pada saat mengirim pesan maka perangkat lunak secara otomatis memasukkan isi pesan pada data pesan terkirim. Seperti pada pesan masuk, pesan keluar mempunyai antar muka yang hampir sama. Antar muka dari pesan keluar

ditunjukkan pada gambar 4.13. Proses pemanggilan data pesan keluar dilakukan kelas Sent_item.java ini adalah sebagai berikut:

```
Key key = KeyManager.getInstance().getLookup().get(number);

String pruned = msg.substring(1);
byte[] clearText = null;

if (key.key != null) {
    try {
            test = new TripleDES(key.key);
            tampilpesan = test.decrypt(pruned);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
Send_item_msg[pos] = tampilpesan ;
pos += 1;
```




Gambar 4.13 Antar Muka Pesan Keluar

# BAB V
# PENGUJIAN DAN ANALISIS

Pada bab V ini akan dijelaskan strategi pengujian, hasil pengujian dan analisis terhadap hasil dari pengujian yang dilakukan.

## 5.1 Strategi Pengujian

Pengujian terhadap perangkat lunak dibagi menjadi dua, yaitu pengujian fungsionalitas perangkat lunak dan pengujian pembengkakan *ciphertext* pesan. Dalam pengujian, data yang digunakan adalah data kunci dan data pesan SMS. Penjelasan lebih lengkap mengenai data kunci yang akan diuji dapat dilihat pada Tabel 5.1. Sedangkan data pesan yang akan di uji dapat dilihat pada Tabel 5.2 dan Tabel 5.3.

Tabel 5.1 Tabel Data Kunci

| No | Pembangkitan kunci | Session key | Hasil enkripsi kunci |
|---|---|---|---|
| 1 | | 33183382<br>37975550<br>48656575 | 8iVnIiZVix+aBoEyNs4mGnZ18WbvKng8YiZv5JEYCss= |
| 2 | | 36110554<br>37975550<br>48656575 | fXHz4hXlso/jzkAOBBeQcnZ18WbvKng8YiZv5JEYCss= |

| | | | |
|---|---|---|---|
| 3 | | 36323612<br>37975550<br>48656575 | kcu76n1C51j98GdwIZJoq3Z18WbvKng8YiZv5JEYCss= |
| 4 | | 62442987<br>37975550<br>48656575 | rJ4UBy7ijzd++yLFJLS9pnZ18WbvKng8YiZv5JEYCss= |
| 5 | | 99220862<br>37975550<br>48656575 | AcFbK1L0vGIGQiJ+n8djgXZ18WbvKng8YiZv5JEYCss= |
| 6 | | 30416968<br>48656575<br>37975550 | PC/U5+jZuw0BahmclaMtaU+BMzJQj85NaHyQ012IIog= |
| 7 | | 55425557<br>48656575<br>37975550 | /z2lprWoxUzYHqg/pEJiVU+BMzJQj85NaHyQ012IIog= |

| | | | |
|---|---|---|---|
| 8 | | 61694027<br>48656575<br>37975550 | EF6wayAGrHUi5HQ43vf1b0+BMzJQj85NaHyQ012IIog= |
| 9 | | 70364668<br>48656575<br>37975550 | oKQM7Gg8+S9wOkk7VoWqmk+BMzJQj85NaHyQ012IIog= |
| 10 | | 62473960<br>48656575<br>37975550 | IbV1/BCI7xwo/PDZ6V56/0+BMzJQj85NaHyQ012IIog= |

Tabel 5.2 Daftar Isi Pesan SMS pengujian fungsionalitas

| No | Isi pesan | Jumlah karakter |
|---|---|---|
| 1 | T | 1 |
| 2 | Tes | 3 |
| 3 | Testing | 7 |
| 4 | Testing. | 8 |
| 5 | Arnes Dusimar S. | 16 |
| 6 | TesenkripsiSMSdenganalgoritmaTDES | 33 |

| | | |
|---|---|---|
| 7 | Tes enkripsi SMS dengan algoritma TDES | 38 |
| 8 | Enkripsi SMS dengan algoritma TDES Tes | 38 |
| 9 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel | 68 |
| 10 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel berbasis android. | 86 |

Tabel 5.3 Daftar Isi Pesan SMS pengujian pembengkakan data

| No. | Plaintext | Jumlah karakter |
|---|---|---|
| 1 | K | 1 |
| 2 | Ko | 2 |
| 3 | Kom | 3 |
| 4 | Komu | 4 |
| 5 | Komun | 5 |
| 6 | Komuni | 6 |
| 7 | Komunik | 7 |
| 8 | Komunika | 8 |
| 9 | Komunikas | 9 |
| 10 | Komunikasi | 10 |
| 11 | KomunikasiY | 11 |

| | | |
|---|---|---|
| 12 | KomunikasiYa | 12 |
| 13 | KomunikasiYan | 13 |
| 14 | KomunikasiYang | 14 |
| 15 | KomunikasiYangS | 15 |
| 16 | KomunikasiYangSe | 16 |
| 17 | KomunikasiYangSer | 17 |
| 18 | KomunikasiYangSeri | 18 |
| 19 | KomunikasiYangSerin | 19 |
| 20 | KomunikasiYangSering | 20 |
| 21 | KomunikasiYangSeringD | 21 |
| 22 | KomunikasiYangSeringDi | 22 |
| 23 | KomunikasiYangSeringDil | 23 |
| 24 | KomunikasiYangSeringDila | 24 |
| 25 | KomunikasiYangSeringDilak | 25 |
| 26 | KomunikasiYangSeringDilaku | 26 |
| 27 | KomunikasiYangSeringDilakuk | 27 |
| 28 | KomunikasiYangSeringDilakuka | 28 |
| 29 | KomunikasiYangSeringDilakukan | 29 |
| 30 | KomunikasiYangSeringDilakukanM | 30 |
| 31 | KomunikasiYangSeringDilakukanMe | 31 |

| | | |
|---|---|---|
| 32 | KomunikasiYangSeringDilakukanMel | 32 |
| 33 | KomunikasiYangSeringDilakukanMela | 33 |
| 34 | KomunikasiYangSeringDilakukanMelal | 34 |
| 35 | KomunikasiYangSeringDilakukanMelalu | 35 |
| 36 | KomunikasiYangSeringDilakukanMelalui | 36 |
| 37 | KomunikasiYangSeringDilakukanMelaluiP | 37 |
| 38 | KomunikasiYangSeringDilakukanMelaluiPo | 38 |
| 39 | KomunikasiYangSeringDilakukanMelaluiPon | 39 |
| 40 | KomunikasiYangSeringDilakukanMelaluiPons | 40 |
| 41 | KomunikasiYangSeringDilakukanMelaluiPonse | 41 |
| 42 | KomunikasiYangSeringDilakukanMelaluiPonsel | 42 |
| 43 | KomunikasiYangSeringDilakukanMelaluiPonselY | 43 |
| 44 | KomunikasiYangSeringDilakukanMelaluiPonselYa | 44 |
| 45 | KomunikasiYangSeringDilakukanMelaluiPonselYai | 45 |
| 46 | KomunikasiYangSeringDilakukanMelaluiPonselYait | 46 |
| 47 | KomunikasiYangSeringDilakukanMelaluiPonselYaitu | 47 |
| 48 | KomunikasiYangSeringDilakukanMelaluiPonselYaituS | 48 |
| 49 | KomunikasiYangSeringDilakukanMelaluiPonselYaituSM | 49 |
| 50 | KomunikasiYangSeringDilakukanMelaluiPonselYaituSMS | 50 |

## 5.2 Hasil Pengujian

Berdasarkan perancangan pengujian yang telah dijelaskan pada bab 3, maka hasil pengujian akan dijelaskan menjadi beberapa sub bab pengujian, yaitu hasil pengujian fungsionalitas perangkat lunak dan hasil pengujian pembengkakan data hasil perangkat lunak.

### 5.2.1. Hasil Pengujian Fungsionalitas Perangkat Lunak

Pengujian fungsionalitas perangkat lunak terbagi menjadi tiga yaitu proses pembangkitan *session key* serta pertukaran kunci, pengiriman pesan dan pembacaan pesan. Nomor ponsel yang digunakan adalah 081937975550 dan 085648656575.

Tabel 5.4 Hasil Pengujian Fungsionalitas

| No | Pembangkitan Kunci | Session key | Hasil enkripsi kunci | Hasil scan kunci | Session key | Hasil dekripsi kunci |
|---|---|---|---|---|---|---|
| 1 | | 3318<br>3382<br>3797<br>5550<br>4865<br>6575 | 8iVnIiZVix+aBoEyNs4mGnZ18WbvKng8YiZv5JEYCss= | | 3318<br>3382<br>3797<br>5550<br>4865<br>6575 | MzMxODMzODIzNzk3NTU1MDQ4NjU2NTc1 |
| 2 | | 3611<br>0554<br>3797<br>5550<br>4865<br>6575 | fXHz4hXIso/jzkAOBBeQcnZ18WbvKng8YiZv5JEYCss= | | 3611<br>0554<br>3797<br>5550<br>4865<br>6575 | MzYxMTA1NTQzNzk3NTU1MDQ4NjU2NTc1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | | 3632<br>3612<br>3797<br>5550<br>4865<br>6575 | kcu76n1C5<br>1j98GdwIZJ<br>oq3Z18Wb<br>vKng8YiZv5<br>JEYCss= | | 3632<br>3612<br>3797<br>5550<br>4865<br>6575 | MzYzMjM2<br>MTIzNzk3NT<br>U1MDQ4Nj<br>U2NTc1 |
| 4 | | 6244<br>2987<br>3797<br>5550<br>4865<br>6575 | rJ4UBy7ijzd<br>++yLFJLS9p<br>nZ18WbvK<br>ng8YiZv5JE<br>YCss= | | 6244<br>2987<br>3797<br>5550<br>4865<br>6575 | NjI0NDI5OD<br>czNzk3NTU1<br>MDQ4NjU2<br>NTc1 |
| 5 | | 9922<br>0862<br>3797<br>5550<br>4865<br>6575 | AcFbK1L0v<br>GlGQiJ+n8d<br>jgXZ18Wbv<br>Kng8YiZv5J<br>EYCss= | | 9922<br>0862<br>3797<br>5550<br>4865<br>6575 | OTkyMjA4Nj<br>IzNzk3NTU1<br>MDQ4NjU2<br>NTc1 |
| 6 | | 3041<br>6968<br>4865<br>6575<br>3797<br>5550 | PC/U5+jZu<br>w0Bahmcla<br>MtaU+BMz<br>JQj85NaHy<br>Q012IIog= | | 3041<br>6968<br>4865<br>6575<br>3797<br>5550 | MzA0MTY5<br>Njg0ODY1Nj<br>U3NTM3OTc<br>1NTUw |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | | 5542<br>5557<br>4865<br>6575<br>3797<br>5550 | /z2IprWoxUzYHqg/pEJiVU+BMzJQj85NaHyQ012Ilog= | | 5542<br>5557<br>4865<br>6575<br>3797<br>5550 | NTU0MjU1NTc0ODY1NjU3NTM3OTc1NTUw |
| 8 | | 6169<br>4027<br>4865<br>6575<br>3797<br>5550 | EF6wayAGrHUi5HQ43vf1b0+BMzJQj85NaHyQ012Ilog= | | 6169<br>4027<br>4865<br>6575<br>3797<br>5550 | NjE2OTQwMjc0ODY1NjU3NTM3OTc1NTUw |
| 9 | | 7036<br>4668<br>4865<br>6575<br>3797<br>5550 | oKQM7Gg8+S9wOkk7VoWqmk+BMzJQj85NaHyQ012Ilog= | | 7036<br>4668<br>4865<br>6575<br>3797<br>5550 | NzAzNjQ2Njg0ODY1NjU3NTM3OTc1NTUw |
| 10 | | 6247<br>3960<br>4865<br>6575<br>3797<br>5550 | IbV1/BCI7xwo/PDZ6V56/0+BMzJQj85NaHyQ012Ilog= | | 6247<br>3960<br>4865<br>6575<br>3797<br>5550 | NjI0NzM5NjA0ODY1NjU3NTM3OTc1NTUw |

Dengan bilangan acak dari sistem adalah 14661404 maka kunci yang dipertukarakan melalui *Barcode Scanner* adalah sebagai berikut MTQ2NjE0MDQzNzk3NTU1MDQ4NjU2NTc1 atau dalam ASCII 14661404379755504865657 5. Hasil pengujian pengiriman pesan menggunakan perangkat lunak ditunjukkan Tabel 5.5 :

Tabel 5.5 Hasil Pengujian Pengiriman Dan Hasil *Ciphertext* Pesan

| No | Isi pesan (Pesan keluar) | Jumlah karakter | Isi pesan (Pesan masuk) | Jumlah karakter |
|---|---|---|---|---|
| 1 | T | 1 | _2MEPG5gjTnE= | 13 |
| 2 | Tes | 3 | _UJcaJ7pWiKo= | 13 |
| 3 | Testing | 7 | _vYYf/1Vcy68= | 13 |
| 4 | Testing. | 8 | _vHMWWaf9Peb/nROCWy9nug== | 25 |
| 5 | Arnes Dusimar S. | 16 | _ZNhvruvEsMaEF+ufOH8O+P+dE4JbL2e6 | 33 |
| 6 | TesenkripsiSMSdengan algoritmaTDES | 33 | _23wA+WFEsNMAOOHZRTj15FYT/YxGyTOmiByBK5dCF4YKW3T4a8wL+g== | 57 |
| 7 | Tes enkripsi SMS dengan algoritma TDES | 38 | _JOmVLl6RP6gxuK6Mc4lyCgw3DfcWa7bNXhh0HvizBxrMFTJBbi7aCg== | 57 |
| 8 | Enkripsi SMS dengan algoritma TDES Tes | 38 | _AX4oz6NeAgcfpUCRnCNrjuNZV0Ll+eNhV8nw+o04/labEQrMXnmyAg== | 57 |

| | | | | |
|---|---|---|---|---|
| 9 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel | 68 | _VIqEjSAKgY8mphjhdtC4LYb+rV7vCVR/joP5hb/CovQf1bzauilL1dpwwhc8A6V5nCzH+cRhK5DM 1nGCFHpP4KhU5tselvZ6 | 97 |
| 10 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel berbasis android. | 86 | _VIqEjSAKgY8mphjhdtC4LYb+rV7vCVR/joP5hb/CovQf1bzauilL1dpwwhc8A6V5nCzH+cRhK5DM 1nGCFHpP4GmUByQqQdntZcJ2pYoYc4Tmv6gPkq62zQ== | 121 |

Dengan bilangan acak dari sistem adalah 14661404 maka kunci yang digunakan untuk melakukan proses enkripsi adalah MTQ2NjE0MDQzNzk3NTU1MDQ4NjU2NTc1 atau dalam ASCII 146614043797555048656575. Hasil pengujian pembacaan pesan menggunakan perangkat lunak dengan kunci dekripsi yang sama dengan kunci enkripsi ditunjukkan tabel 5.6 :

Tabel 5.6 Hasil Pembacaan Pesan SMS dengan Kunci Asli

| No | Isi pesan (Pesan keluar) | Jumlah karakter | Isi pesan (Pesan masuk) | Jumlah karakter |
|---|---|---|---|---|
| 1 | T | 1 | T | 1 |
| 2 | Tes | 3 | Tes | 3 |
| 3 | Testing | 7 | Testing | 7 |

| | | | | |
|---|---|---|---|---|
| 4 | Testing. | 8 | Testing. | 8 |
| 5 | Arnes Dusimar S. | 16 | Arnes Dusimar S. | 16 |
| 6 | TesenkripsiSMSdengan algoritmaTDES | 33 | TesenkripsiSMSdengan algoritmaTDES | 33 |
| 7 | Tes enkripsi SMS dengan algoritma TDES | 38 | Tes enkripsi SMS dengan algoritma TDES | 38 |
| 8 | Enkripsi SMS dengan algoritma TDES Tes | 38 | Enkripsi SMS dengan algoritma TDES Tes | 38 |
| 9 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel | 68 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel | 68 |
| 10 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel berbasis android. | 86 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel berbasis android. | 86 |

Pengujian selanjutnya adalah dengan meneruskan isi pesan ke ponsel lain dengan nomor Dengan bilangan acak dari sistem adalah 14661404 maka kunci yang digunakan untuk proses enkripsi adalah MTQ2NjE0MDQzNzk3NTU1MDQ4NjU2NTc1 atau dalam ASCII 14661404379755504865657 5. Sedangkan kunci dekripsi yang digunakan adalah MTIzNDU2NzgzNzk3NTU1MDMzOTQ3MDEx atau dalam ASCII karakter 14661404855566334865657 5. Hasil pengujian pembacaan pesan menggunakan perangkat lunak dengan kunci dekripsi yang berbeda dengan kunci enkripsi ditunjukkan tabel 5.7 :

Tabel 5.7 Hasil Pengujian Pembacaan Pesan SMS Dengan Kunci Palsu

| No | Isi pesan (Pesan keluar) | Jumlah karakter | Isi pesan (Pesan masuk) | Jumlah karakter |
|---|---|---|---|---|
| 1 | T | 1 | _2MEPG5gjTnE= | 13 |
| 2 | Tes | 3 | _UJcaJ7pWiKo= | 13 |
| 3 | Testing | 7 | _vYYf/1Vcy68= | 13 |
| 4 | Testing. | 8 | _vHMWWaf9Peb/nROCWy9nug== | 25 |
| 5 | Arnes Dusimar S. | 16 | _ZNhvruvEsMaEF+ufOH8O+P+dE4JbL2e6 | 33 |
| 6 | TesenkripsiSMSdengan algoritmaTDES | 33 | _23wA+WFEsNMAOOHZRTj15FYT/YxGyTOmiByBK5dCF4YKW3T4a8wL+g== | 57 |
| 7 | Tes enkripsi SMS dengan algoritma TDES | 38 | _JOmVLl6RP6gxuK6Mc4lyCgw3DfcWa7bNXhh0HvizBxrMFTJBbi7aCg== | 57 |
| 8 | Enkripsi SMS dengan algoritma TDES Tes | 38 | _AX4oz6NeAgcfpUCRnCNrjuNZV0Ll+eNhV8nw+o04/labEQrMXnmyAg== | 57 |
| 9 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel | 68 | _VIqEjSAKgY8mphjhdtC4LYb+rV7vCVR/joP5hb/CovQf1bzauilL1dpwwhc8A6V5nCzH+cRhK5DM 1nGCFHpP4KhU5tselvZ6 | 97 |

| | | | | |
|---|---|---|---|---|
| 10 | Implementasi pengamanan pesan SMS dengan manajemen kunci pada ponsel berbasis android. | 86 | _VIqEjSAKgY8mphjhdtC4LYb+rV7vCVR/joP5hb/CovQf1bzauilL1dpwwhc8A6V5nCzH+cRhK5DM 1nGCFHpP4GmUByQqQdntZcJ2pYoYc4Tmv6gPkq62zQ== | 121 |

### 5.2.2. Hasil Pengujian Pembengkakan Data *Ciphertext*

Hasil Pengujian Pembengkakan Data Hasil Perangkat Lunak

Tabel 5.8 Hasil Pengujian Pengiriman Dan Hasil *Ciphertext* Pesan

| No. | Plaintext | Jumlah karakter | Ciphertext | Jumlah karakter | Perbandingan data |
|---|---|---|---|---|---|
| 1 | K | 1 | _4SdMR+gwMbI= | 13 | 1300 |
| 2 | Ko | 2 | _2rBPZPj/DCU= | 13 | 650 |
| 3 | Kom | 3 | _3IbdNKk9gx8= | 13 | 433,3333333 |
| 4 | Komu | 4 | _6S4doFMYjBY= | 13 | 325 |
| 5 | Komun | 5 | _oWcYWxWWs+A= | 13 | 260 |
| 6 | Komuni | 6 | _cfm3lTBi/aw= | 13 | 216,6666667 |
| 7 | Komunik | 7 | _h5pIT8DkPZE= | 13 | 185,7142857 |

| | | | | | |
|---|---|---|---|---|---|
| 8 | Komunika | 8 | _9qGuz7FKdKn/nROCWy9nug== | 25 | 312,5 |
| 9 | Komunikas | 9 | _9qGuz7FKdKnb4tE54gXruA== | 25 | 277,7777778 |
| 10 | Komunikasi | 10 | _9qGuz7FKdKmQfCANS+wgyw== | 25 | 250 |
| 11 | KomunikasiY | 11 | _9qGuz7FKdKlibbNZ9yrGoQ== | 25 | 227,2727273 |
| 12 | KomunikasiYa | 12 | _9qGuz7FKdKnPDOu7/zF5sg== | 25 | 208,3333333 |
| 13 | KomunikasiYan | 13 | _9qGuz7FKdKkGXAKjEwdTfg== | 25 | 192,3076923 |
| 14 | KomunikasiYang | 14 | _9qGuz7FKdKnXjwi7nfx89A== | 25 | 178,5714286 |
| 15 | KomunikasiYangS | 15 | _9qGuz7FKdKnTaPSq1o651w== | 25 | 166,6666667 |
| 16 | KomunikasiYangSe | 16 | _9qGuz7FKdKlrlQ+fnWsjhP+dE4JbL2e6 | 33 | 206,25 |
| 17 | KomunikasiYangSer | 17 | _9qGuz7FKdKlrlQ+fnWsjhBaRD+QB7mxQ | 33 | 194,1176471 |
| 18 | KomunikasiYangSeri | 18 | _9qGuz7FKdKlrlQ+fnWsjhEFTtjsTtC+5 | 33 | 183,3333333 |

| | | | | | |
|---|---|---|---|---|---|
| 19 | KomunikasiYangSerin | 19 | _9qGuz7FKdKlrlQ+fnWsjhIgosQd/tvzL | 33 | 173,6842105 |
| 20 | KomunikasiYangSering | 20 | _9qGuz7FKdKlrlQ+fnWsjhA9nGTZab5CW | 33 | 165 |
| 21 | KomunikasiYangSeringD | 21 | _9qGuz7FKdKlrlQ+fnWsjhDX0CYGjvAiM | 33 | 157,1428571 |
| 22 | KomunikasiYangSeringDi | 22 | _9qGuz7FKdKlrlQ+fnWsjhIDbZuTYVlj/ | 33 | 150 |
| 23 | KomunikasiYangSeringDil | 23 | _9qGuz7FKdKlrlQ+fnWsjhHg9CClUkpxS | 33 | 143,4782609 |
| 24 | KomunikasiYangSeringDila | 24 | _9qGuz7FKdKlrlQ++fnWsjhPDFU++1PMLt/50TglsvZ7o= | 45 | 187,5 |
| 25 | KomunikasiYangSeringDilak | 25 | _9qGuz7FKdKlrlQ++fnWsjhPDFU++1PMLtDan0dwYAc1k= | 45 | 180 |
| 26 | KomunikasiYangSeringDilaku | 26 | _9qGuz7FKdKlrlQ++fnWsjhPDFU++1PMLtzAszTxGFwCA= | 45 | 173,0769231 |

| | | | | | |
|---|---|---|---|---|---|
| 27 | KomunikasiYangSeringDilakuk | 27 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtTqwHX5GnC9k= | 45 | 166,6666667 |
| 28 | KomunikasiYangSeringDilakuka | 28 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtDkjQpoVEmFw= | 45 | 160,7142857 |
| 29 | KomunikasiYangSeringDilakukan | 29 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtZECdjcoEMnc= | 45 | 155,1724138 |
| 30 | KomunikasiYangSeringDilakukanM | 30 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtf5YqjjdvKDQ= | 45 | 150 |
| 31 | KomunikasiYangSeringDilakukanMe | 31 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLt7NZoFjMy5K0= | 45 | 145,1612903 |
| 32 | KomunikasiYangSeringDilakukanMel | 32 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5Gpf/nROCWy9nug== | 57 | 178,125 |
| 33 | KomunikasiYangSeringDilakukanMela | 33 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5Gpdpgrrp9dg0vg== | 57 | 172,7272727 |

| | | | | | |
|---|---|---|---|---|---|
| 34 | KomunikasiYang SeringDilakukan Melal | 34 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpe cxJhyGC3LGQ== | 57 | 167,647 0588 |
| 35 | KomunikasiYang SeringDilakukan Melalu | 35 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpe 7lqhtKzksQw== | 57 | 162,857 1429 |
| 36 | KomunikasiYang SeringDilakukan Melalui | 36 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf zRwFyOsiDbw== | 57 | 158,333 3333 |
| 37 | KomunikasiYang SeringDilakukan MelaluiP | 37 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpe DrXNJi+ucfQ== | 57 | 154,054 0541 |
| 38 | KomunikasiYang SeringDilakukan MelaluiPo | 38 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf As6suRLHt+Q== | 57 | 150 |
| 39 | KomunikasiYang SeringDilakukan MelaluiPon | 39 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpc KKF1uEn4CTg== | 57 | 146,153 8462 |

| | | | | | |
|---|---|---|---|---|---|
| 40 | KomunikasiYang SeringDilakukan MelaluiPons | 40 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf GtByS0sx3LP+dE4J bL2e6 | 65 | 162,5 |
| 41 | KomunikasiYang SeringDilakukan MelaluiPonse | 41 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf GtByS0sx3LEyGdE +idDvV | 65 | 158,536 5854 |
| 42 | KomunikasiYang SeringDilakukan MelaluiPonsel | 42 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf GtByS0sx3LDI4b62 jcFfO | 65 | 154,761 9048 |
| 43 | KomunikasiYang SeringDilakukan MelaluiPonselY | 43 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf GtByS0sx3LGCtMl pwT8su | 65 | 151,162 7907 |
| 44 | KomunikasiYang SeringDilakukan MelaluiPonselYa | 44 | _9qGuz7FKdKlrlQ+ fnWsjhPDFU++1P MLtugLxBmY5Gpf GtByS0sx3LI/6ANA M54YW | 65 | 147,727 2727 |

| | | | | | |
|---|---|---|---|---|---|
| 45 | KomunikasiYangSeringDilakukanMelaluiPonselYai | 45 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LHnWn76pfy/y | 65 | 144,4444444 |
| 46 | KomunikasiYangSeringDilakukanMelaluiPonselYait | 46 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LEDnprrGsXmE | 65 | 141,3043478 |
| 47 | KomunikasiYangSeringDilakukanMelaluiPonselYaitu | 47 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LENYe4tZG9xK | 65 | 138,2978723 |
| 48 | KomunikasiYangSeringDilakukanMelaluiPonselYaituS | 48 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LMlpGrbzkCgX/50TglsvZ7o= | 77 | 160,4166667 |
| 49 | KomunikasiYangSeringDilakukanMelaluiPonselYaituSM | 49 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LMlpGrbzkCgX27Ts2Yt7rmA= | 77 | 157,1428571 |

| | | | | | |
|---|---|---|---|---|---|
| 50 | KomunikasiYangSeringDilakukanMelaluiPonselYaituSMS | 50 | _9qGuz7FKdKlrlQ+fnWsjhPDFU++1PMLtugLxBmY5GpfGtByS0sx3LMlpGrbzkCgX40Oiwe3Fh4c= | 77 | 154 |

## 5.3. Analisis Hasil Pengujian

Hasil uji yang didapatkan akan di analisis lebih lanjut untuk melihat apakah hasilnya sesuai dengan dengan tujuan yang ingin dicapai dalam penelitian ini. Selain itu akan diambil kesimpulan terhadap hasil uji yang didapat.

### 5.3.1. Analisis Hasil Uji Fungsionalitas Perangkat Lunak

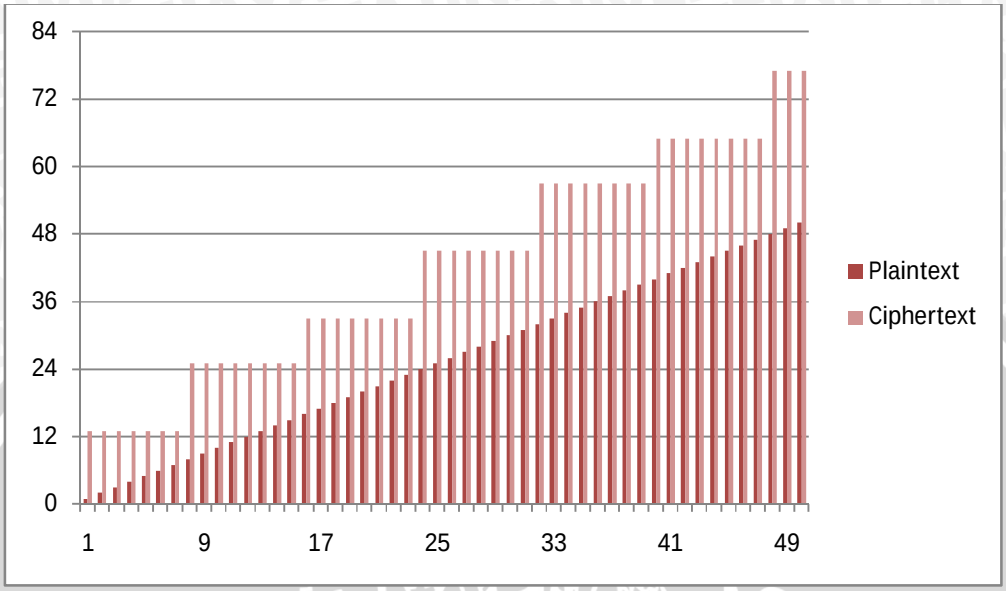
Hasil uji menunjukkan bahwa perangkat lunak dapat membangkitkan dan membaca *session key* dengan baik. Hal ini dibuktikan dengan hasil *session key* yang dibangkitkan oleh sistem sesuai dengan kriteria kunci yang dipaparkan pada Bab 3. Selain itu semua pertukaran kunci dapat berjalan dengan baik dan benar karena semua kunci dapat diterima sesuai dengan keadaan aslinya baik melalui *scan barcode* maupun kirim *ciphertext* kunci melalui SMS.

Perangkat lunak ini juga berhasil melakukan enkripsi dan dekripsi pesan dengan kunci yang sama. Hal ini dibuktikan dengan pesan asli yang dikirim sama dengan hasil pesan yang diterima. Bila kunci berbeda maka perangkat lunak tidak melakukan proses dekripsi pesan, dan pesan yang ditampilkan berupa *ciphertext* yang diterima.

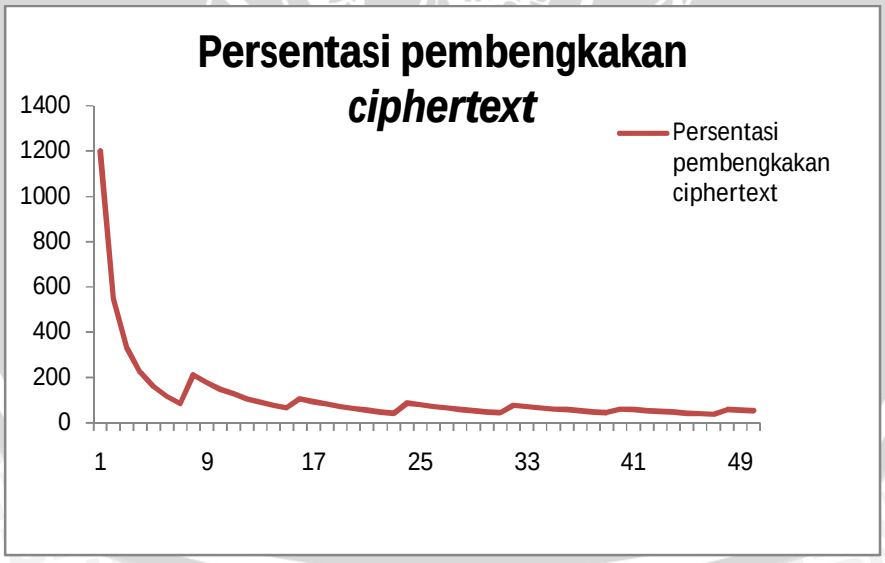
### 5.3.2. Analisis Hasil Uji Pembengkakan *Ciphertext* Perangkat Lunak

Dari hasil uji coba pengiriman dan penerimaan pesan dengan perangkat lunak ini terdapat pembengkakan data. Hal ini dapat dilihat dari lebih besarnya data *ciphertext* daripada pesan asli. Perbandingan *plaintext* dan *ciphertext*

direpresentasikan oleh grafik 5.1. Sedangkan persentasi pembengkakan *ciphertext* pada perangkat lunak ini dapat ditunjukkan oleh grafik 5.2.




Gambar 5.1 Grafik Perbandingan *Plaintext* dan *Ciphertext*




Gambar 5.2 Grafik Persentase Pembengkakan *Ciphertext*

Dari grafik diatas dapat diambil kesimpulan bahwa semua *ciphertext* yang dihasilkan mengalami pembengkakan data. Semakin banyak karakter *plaintext*

maka persentasi pembengkakan data semakin menurun tetapi naik kembali pada tiap interval 8 karakter.

## 5.4. Analisis Umum Hasil Uji

Perangkat lunak yang dihasilkan dinilai telah bekerja dengan baik dan cukup memuaskan meskipun terjadi pembengkakan *ciphertext*. Perangkat lunak dapat melakukan enkripsi dan dekripsi dengan baik. Perangkat lunak juga dapat membangkitkan kunci yang sesuai kriteria. Selain itu, perangkat lunak juga dapat membaca kunci melalui barcode scanner sehingga pertukaran kunci dapat berjalan baik.

# BAB VI
# KESIMPULAN DAN SARAN

## 6.1. Kesimpulan

Setelah menyelesaikan skripsi ini maka disimpulkan :

1. Teknologi enkripsi, dekripsi, dan sistem manajemen kunci kriptografi pesan SMS pada telepon genggam berbasis android dapat diimplementasikan.
2. *Ciphertext* pesan yang dihasilkan dengan kriptografi ini mengalami pembengkakan data yang cukup besar yaitu antara 40% sampai 1200%. Semakin besar data *plaintext* maka pembengkakan data semakin kecil tertapi naik kembali pada tiap interval 8 karakter.
3. Pembangkitan session key yang terdiri dari bilangan acak dan nomor ponsel pengguna dapat meningkatkan keamanan data. Nomor ponsel pengguna dapat digunakan sebagai otentikasi pesan.

## 6.2. Saran

Saran yang dapat penulis berikan setelah mengerjakan skripsi ini adalah sebagai berikut :

1. Agar dapat mengurangi pembengkakan data, maka dibutuhkan kompresi sebelum pesan dikirim.
2. Perlunya metode baru untuk mengambil data nomor ponsel untuk menghilangkan ketergantungan perangkat lunak terhadap aplikasi *Whatsapp*.
3. Agar keamanan pesan dapat lebih ditingkatkan perlu diterapkan algoritma enkripsi kunci asimetris.

# DAFTAR PUSTAKA


[ARI-11] Ariyana, Yoki. 2011. *Advanced Encryption Standard (AES)*. Bandung : PPPPTK IPA Bandung.

[ARI-08] Ariyus, D. 2008. *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi*. Andi Offset. Yogyakarta.

[GUE-10] Gueron, Shay. 2010. *Intel® Advanced Encryption Standard (AES) New Instructions Set*. Mobility Group, Israel Development Center Intel Corporation.

[HER-99] Heriyanto, Tedy.1999. *Pengenalan Kriptografi*. hal 2.

[KRE-09] Kretzschmar, Uli. 2009. *AES128 – A C Implementation for Encryption and Decryption*. Texas Instrument.

[MEN-97] Menezes, A.J., 1997. *Handbook of Applied Cryptography*. CRC Press.

[MUN-04] Munir, Rinaldi. 2004. *Kuliah IF5054 Kriptografi :Tipe dan Mode Algoritma Simetri*. Institut Teknologi Bandung. Bandung.

[MNR-04] Munir, Rinaldi, *Manajemen kunci, Bahan Kuliah IF5054*, pp. 1–2, 2004. Institut Teknologi Bandung. Bandung.

[MUN-06] Munir, R. 2006. *Kriptografi.Edisi ke-1*. Bandung: Informatika.

[KEL-12] Keller, Sharon. 2012. *Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS) : Requirements and Procedures*. NIST Special Publication 800-20.

[RAT-06] Ratih. 2006. *Studi dan Perbandingan Penggunaan Kriptografi Kunci Simetri dan Asimetri pada Telepon Selular*. Institut Teknologi Bandung. Bandung.

[PAU-98] Paul, Gautam, Shubamoy Maitra. 1998 *RC4 Stream Cipher and Its Variants* . CRC Press.Sistem keamanan komputer. Jakarta.

[NIS-99] NIST Computer Security Division (CSD).1999. *FIPS 46-3: The official document describing the DES standard*. Federal Information Processing Standards Publication.