IMPLEMENTASI SISTEM KRIPTOGRAFI DALAM PENGAMANAN DATABASE SMS MENGGUNAKAN STREAM CIPHER RC4

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

Oleh :

NAMA : **MUHAMMAD RIZKY**
NPM : **1414370226**
PROGRAM STUDI : **SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

Muhammad Rizky

ImplementasiSistem Kriptografi Dalam Pengamanan Database Sms Menggunakan Stream Cipher RC4

2019

Penelitian ini membahas tentang pengamanan terhadap database SMS dan penyisipan database SMS ke dalam suatu media dapat diterapkan dengan aplikasi pengamanan yaitu Stream Cipher. Aplikasi Stream Cipher adalah jenis algoritma enkripsi simetrik. Stream Cipher digunakan untuk blok data yang lebih kecil, biasanya ukuran bit. RC4 merupakan salah satu jenis stream cipher, Implementasi algoritma RC4 Untuk mengenkripsi database SMS yang dikirim. Algoritma RC4 merupakan salah satu algoritma kunci simetris berbentuk stream cipher yang memproses unit atau input data, pesan atau informasi pada satu saat. Unit atau data pada umumnya sebuah byte atau bahkan bit (byte dalam hal RC4). Algoritma ini tidak harus menunggu sejumlah input data atauinformasi tertentu. RC4mempunyai sebuah S-Box, S0,S1,……,S255, yang berisi permutasi dari bilangan 0 sampai 255. Dengan dua buah indeks yaitu i dan j di dalam algoritmanya. Proses kerja dari penelitian ini adalah dengan menggunakan Mysql sebagai database server dan PHP sebagai bahasa pemrogramannya.

Kata Kunci : Database, Mysql, RC4, SMS, Stream Cipher.

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan autentikasi entitas. Namun, seiring berkembangannya teknologi, kriptografi ini juga berkembang, perkembangan teknologi ini bisa kita lihat dengan adanya internet yang dapat menghubungkan komputer kita dengan komputer yang lainnya. Dengan adanya perkembangan ini kriptografi sangat dibutuhkan untuk keamanan data yang dikirim kepada penerima data lain.

Ada empat tujuan pokok dari kriptografi. Kerahasiaan *(confidentiality)* dimana kriptografi digunakan untuk menjaga isi dari informasi dari pihak manapun kecuali pemilik otoritas yang mempunyai kunci rahasia atau sandi. Kerahasiaan bisa dijaga dengan melakukan enkripsi. Keutuhan *(integrity)* yang bersangkutan dengan keamanan dari perubahan data secara tidak sah. Untuk menjaga keutuhan data, maka sistem yang digunakan haruslah mempunyai kemampuan mendeteksi manipulasi data yang dilakukan pihak ketiga.

Kegunaan lainnya dari kriptografi ialah autentikasi yang bersangkutan dengan identifikasi atau pengenalan. Dua pihak yang saling berkomunikasi haruslah saling kenal. Informasi yang dikirimkan haruslah diautentifikasi keaslian datanya, waktu pengiriman dan lain-lain. *Non-repudiation* merupakan usaha untuk mencegah terjadinya penyangkalan pengiriman data.

Tingkat keamanan pada sistem kriptografi bisa kita terapkan pada media komunikasi saat ini. Salah satunya ialah SMS *(Short Message Service).* Sebuah layanan pada telepon genggam untuk mengirim dan menerima pesan-pesan pendek. Keamanan pada SMS juga menjadi perhatian penting yang perlu ditambahkan dalam penggunaannya. Sekarang ini sering juga terjadi pembobolan SMS dari pihak tidak bertanggung jawab. Pengamanan terhadap pesan asli dan penyisipan pesan SMS ke dalam suatu media dapat diterapkan dengan aplikasi pengamanan yaitu *Stream Cipher*.

Aplikasi *Stream Cipher* adalah jenis algoritma enkripsi simetrik. *Stream Cipher* digunakan untuk blok data yang lebih kecil, biasanya ukuran bit. Satu *Stream Cipher* menghasilkan suatu *keystream* (satuan barisan bit yang digunakan sebagai kunci). Proses enksripsi dicapai dengan menyatukan *keystream* dengan *plaintext* biasanya dengan operasi *bitwise XOR*. Pembentukan *keystream* bisa dibuat independen dengan *plaintext* dan *chipertext. Stream cipher* pada umumnya berkaitan dengan sifat *one-time pad*. Suatu *one-time pad* kadang-kadang disebut *vernam chiper*, dengan sebuah *string* dari bit yang murni secara acak.

RC4 merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau input data, pesan atau informasi data pada satu saat. Unit atau data pada umumnya adalah sebuah *byte*. Dengan cara enkripsi ini dapat dilaksanakan pada panjang yang variable. Algoritma ini tidak harus menunggu sejumlah inputan data, pesan atau informasi tertentu sebelum diproses, atau menambahkan byte tambahan untuk mengenkrip.

RC4 mempunyai sebuah *S-Box* yang berisikan permutasi dari bilangan 0 sampai 255. Menggunakan dua buah indeks yaitu i dan j di dalam algoritmanya. Indeks i digunakan untuk memastikan bahwa suatu elemen berubah, sedangkan indeks j memastikan bahwa elemen berubah secara random.

Dengan latar belakang yang tertera, dalam perancangan dan penulisan skripsi ini, penulis berinisiatif memberikan judul "**Implementasi Sistem Kriptografi Dalam Pengamanan Database Short Message Service Menggunakan Stream Cipher RC4**".

1.2 Rumusan Masalah

Dari latar belakang diatas penulis dapat merumuskan masalah yang ada antara lain :

1. Bagaimana merancang sistem kriptografi dalam database SMS *(short message service)* dengan mengunakan *stream cipher RC4* ?
2. Bagaimana menghindari penyadapan yang dilakukan oleh pihak tidak bertanggung jawab dalam penggunaan SMS ?
3. Apakah tingkat keamanan database SMS terjaga dengan baik menggunakan *stream cipher* secara signifikan ?

1.3 Batasan Masalah

Berdasarkan latar belakang masalah diatas, maka pokok permasalahan penelitian ini dibatasi dalam hal-hal sebagai berikut :

1. Sistem keamanan kriptografi menjadi pokok pembahasan pada penelitian.
2. Proses kerja dari penelitian ini adalah dengan menggunakan *Mysql* sebagai *database server* dan PHP sebagai bahasa pemrogramannya.
3. Proses enkripsi menggunakan algoritma *RC4* dengan panjang kunci yang digunakan oleh *RC4* adalah 1 *byte* hingga 256 *byte* dan digunakan untuk menginsialisasikan tabel sepanjang 256 *byte*.

1.4 Tujuan Penelitian

Adapun tujuan yang akan dicapai dalam penelitian ini, antara lain:

1. Untuk merancang sistem kriptografi dalam *database* SMS dengan menggunakan *stream cipher RC4.*
2. Untuk menghindari penyadapan yang dilakukan oleh pihak tidak bertanggung jawab dalam penggunaan SMS.
3. Untuk merancang tingkat keamanan *database* yang aman dan baik dengan menggunakan metode *stream cipher RC4.*

1.5 Manfaat Penelitian

Manfaat dari perancangan dan penelitian ini dapat dilihat sebagai berikut :

1. Agar dapat memberikan kenyamanan dalam hal keamanan *database* pada saat mengirimkan *SMS.*
2. Hasil penelitian diharapkan dapat menghindari dari bahayanya penyadapan yang dilakukan oleh pihak tidak bertanggung jawab.
3. Dengan perancangan keamananan *database SMS* menggunakan *stream cipher RC4* diharapkan penggunaannya dapat diterapkan dengan baik.

BAB II

LANDASAN TEORI

2.1 Pengertian Data dan Informasi

Data dapat didefenisikan sebagai bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang tidak acak, yang menunjunkan jumlah, tindakan, atau hal. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan sebagai file dalam basis data. Data menjadi bahan dalam suatu proses pengolahan data. Oleh karena itu, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut. Contoh data adalah catatan identitas pegawai, catatan transaksi pembelian (Zefriyeni dan Santoso, 2015).

Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dilihat akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Untuk memperoleh informasi, diperlukan data yang akan diolah dan unit pengolah (Sutanta, 2011).

Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi bentuk yang berguna bagi penerimanya (Tata Sutabri, 2012).

2.2 Komponen Sistem Informasi

Dalam arti yang luas sistem informasi dapat dipahami sebagai sekumpulan susbsistem yang saling berhubungan. Berdasarkan komponen fisik penyusunnya, sistem informasi terdiri atas komponen berikut (Sutanta, 2003).

2.2.1 Perangkat keras *(hardware)*

Perangkat keras dalam sistem informasi meliputi perangkat yang digunakan oleh sistem komputer untuk masuk dan keluaran *(input/output device), memory, modem, processor* dan perifera lainnya.

2.2.2 Perangkat lunak *(software)*

Perangkat lunak dalam sistem informasi adalah berupa program-program komputer yang meliputi sistem operasi *(operating system)* bahasa pemrograman *(programming language)* dan program aplikasi lain

2.2.3 Berkas basis data *(file)*

Berkas merupakan sekumpulan data dalam basis data yang disimpan dengan cara-cara tertentu sehingga dapat digunakan kembali dengan mudah dan cepat.

2.2.4 Prosedur *(procedure)*

Prosedur meliputi prosedur pengoperasian untuk sistem informasi manual, dan dokumen-dokumen yang memuat aturan-aturan yang berhubungan dengan sistem informasi dan lainnya.

2.2.5 Manusia *(brainware)*

Manusia yang terlibat dalam suatu sistem informasi meliputi operator, *programmer, system analyst*, manajer informasi, manajer pada tingkat

operasional, manajer pada tingkat strategis, teknisi administrator basis data *(DBA)*, serta individu lain yang terlibat didalamnya.

2.3 Keamanan Data

Keamanan data *(data security)* menurut Sutanta, merupakan aspek kritis dalam basis data. Prinsip dasar dari keamanan data dalam basis data adalah bahwa data-data dalam basis data merupakan sumber informasi yang bersifat sangat penting dan rahasia. Oleh karena itu, data-data tersebut harus dijaga dari berbagai hal yang memungkinkan dapat merusak data. Aspek keamanan data meliputi (Martin, 1975).

1. *Recovery*

   Adalah suatu proses menggunakan kembali data dari media penyimpanan cadangan untuk mengembalikan data pada kondisi yang benar karena terjadi kerusakan/kehilangan.

2. *Integrity*

   Berkaitan dengan unjuk kerja sistem untuk dapat menjaga data-data dalam basis data agar selalu berada dalam kondisi yang benar, konsisten dan selalu tersedia.

3. *Concurency*

   Berkaitan dengan mekanisme pengendalian basis data saat digunakan oleh beberapa pemakai pemakai secara bersamaan agar terhindar dari kesalahan akibat beberapa transaksi berbeda dilakukan secara bersamaan.

4. *Privacy*

   Yaitu sebagai pembatasan kewenangan akses data untuk mencegah dan melindungi data dari penggunaan oleh pengguna yang tidak berwenang dan mengubah secara tidak diketahui.

5. *Security*

   Adalah suatu mekanisme sistem untuk mencegah dan melindungi data kehilangan akibat kerusakan pada fisik media penyimpanan, kebakaran, banjir, badai, huru-hara, dan lain-lain.

2.4 Ancaman Keamanan Data

Banyak terjadinya pertukaran informasi setiap detik di internet. Banyak terjadinya pencurian dari informasi itu sendiri oleh pihak tidak bertanggung jawab. Ancaman keamanan data yang terjadi terhadap informasi adalah (Doni Ariyus, 2006).

1. *Interruption*

   Merupakan suatu ancaman terhadap avaibability, keamanan informasi data yang ada dalam sistem komputer dirusak, dihapus, sehingga jika data informasi tersebut dibutuhkan tidak ada lagi.

2. *Interception*

   Merupakan ancaman terhadap kerahasiaan. Informasi yang ada disadap atau orang yang tidak berhak mendapat akses ke komputer dimana informasi tersebut disimpan.

3. *Modification*

   Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalulintas informasi yang sedang dikirim dan dirubah sesuai dengan keinginan orang tersebut.

4. *Fabrication*

   Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru (memalsukan) suatu informasi yang ada sehingga orang yang menerima informasi tersebut menyangka informasi itu bersal dari orang yang dikehendaki oleh sipenerima informasi itu.

2.5 Sejarah Kriptografi

Kriptografi mempunyai sejarah yang menarik dan panjang. Kriptografi sudah digunakan 4000 tahun yang lalu, diperkenalkan oleh orang-orang Mesir lewat hieroglyph. Jenis tulisan ini bukanlah bentuk standar untuk menulis pesan. Dikisahkan, pada zaman Romawi kuno, pada suatu saat Julius Caesar ingin mengirimkan pesan rahasia kepada seorang jendral di medan perang yang dikirim lewat kurir, Julius Caesar tidak ingin pesan rahasia tersebut terbuka dijalan. Kemudian ia memikirkan bagaimana mengatasinya, kemudian ia mengacak pesan tersebut menjadi suatu pesan yang tidak dapat dipahami oleh siapapun terkecuali jendralnya saja. Sang jendral telah diberitahu sebelumnya bagaimana cara membaca pesan teracak tersebut. Yang dilakukan Julius Caesar adalah mengganti semua susunan alphabet dari a,b,c, yaitu a menjadi d, b menjadi e, c manjadi f dan seterusnya.

Dari ilustrasi tersebut, beberapa istilah kriptografi dipergunakan untuk menandai aktivitas-aktivitas rahasia dalam mengirim pesan. Apa yang dilakukan Julius Caesar yang mengacak pesan, disebut sebagai enkripsi. Pada saat jendral merapikan pesan yang teracak itu, disebut dengan dekripsi. Pesan awal yang belum diacak disebut *plaintext*, sedangkan pesan yang sudah diacak disebut *ciphertext.*

2.6 Pengertian Kriptografi

Kriptografi berasal dari bahasa Yunani, *Crypto* berarti rahasia dan *graphia* berate tulisan. Menurut terminologinya *kriptografi* adalah ilmu dan seni untuk menjaga keamanan pesan ketika dikirim dari suatu tempat ke tempat lain. Ilmu dan seni ini sebenarnya sudah digunakan sejak dahulu kala di bidang militer dan agen rahasia. Dan sampai sekarang para ahli masih terus mengembangkan dan meneliti penggunaannya (Doni Ariyus, 2008).

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematis yang berhubungan dengan aspek keamanan informasi seperti keabsahan, integritas data, serta autentikasi data. Kriptografi tidak hanya memberikan keamanan informasi saja, namun lebih kearah teknik-tekniknya.

Kriptografi mempunyai 2 bagian yang penting, yaitu enkripsi dan deskripsi. Enkripsi adalah proses penyandian dari yang asli menjadi yang tidak dapat diartikan seperti pesan aslinya. Deskripsi sendiri berarti merubah pesan yang sudah disandikan menjadi pesan yang asli. Pesan asli biasanya disebut *plaintext*, sedangkan pesan yang sudah disandikan disebut *chipertext*. Adapun

algoritma matematis yang digunakan pada proses enkripsi yakni disebut chipper dan sistem yang memanfaatkan kriptografi untuk mengamankan sistem informasi disebut kriptosistem.

Dalam perkembangannya, kriptografi juga digunakan untuk mengedintifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital.

2.6.1 Komponen Kriptografi

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, seperti dibawah ini menurut (Doni Ariyus, 2008).

1. Enkripsi

   Merupakan hal yang sangat penting dalam kriptografi, merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaanya. Pesan asli disebut *plaintext* (teks-biasa), yang diubah menjadi kode-kode yang tidak dapat dimengerti. Enkripsi bisa diartika dengan *cipher* atau kode. Sama halnya dengan tidak mengerti sebuah kata maka kita akan melihatnya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk teks-biasa ke bentuk teks-kode kita gunakan algoritma yang dapat mengkodekan data yang kita inginkan.

2. Dekripsi

   Merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan kembali ke bentuk asalnya. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunkan untuk enkripsi.

3. Kunci

   Kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yaitu rahasia *(private key)* dan kunci umum *(public key).*

4. *Ciphertext*

   Merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).

5. *Plaintext*

   Sering disebut dengan *cleartext.* Teks-asli atau teks-biasa ini merupakan pesan yang dituliskan atau diketik yang meliliki makna. Teks-asli inilah yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext.*

6. Pesan

   Dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman (kertas, *storage*, dsb).

7. *Cryptanalysis*

   Kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar. Jika suatu teks-kode berhasil diubah menjadi teks-asli tanpa menggunkan

kunci 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 kriptografi.

2.6.2 Algoritma Kriptografi

Kata algorism mempunyai arti proses perhitungan dalam bahasa Arab. Algoritma berasal dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad Ibnu Musa al-Khuwariziwi (al-Khuwarizwi dibaca oleh oranhg barat sebagai algorism). Kata *algorism* lambat laun berubah menjadi *algorithm.*

Defenisi terminologi algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang yang tidak berhak mengetahui pesan itu. Algoritma kriptografi terdiri dari tiga fungsi dasar yaitu enkripsi, dekripsi, dan kunci (Doni Ariyus, 2008).

Keamanan dari algoritma kriptografi tergantung pada bagaimana algoritma itu bekerja. Oleh sebab itu algoritmna semacam ini disebut dengan algoritma terbatas. Algoritma terbatas merupakan algoritma yang dipakai sekelompok orang untuk merahasiakan pesan yang meraka kirim. Jika salah satu dari anggota itu keluar dari kelompoknya maka harus diganti dengan yang baru.

Keamanan dari kriptografi modern didapat dengan merahasiakan kunci yang dimiliki dari orang lain, tanpa harus merahasiakan algoritma itu sendiri. Kunci memiliki fungsi yang sama dengan password. Jika keseluruhan kunci

algoritma tergantung pada kunci yang dipakai maka algoritma itu bisa dipublikasikan dan diteliti orang lain (Darma, 2017).

2.6.3 Macam-macam Kriptografi

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya:

1. Algoritma Simetri

   Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Keamanan dari pesan yang menggunkan algoritma ini tergantung pada kunci. Algoritma yang memakai kunci simetri diantaranya adalah:

   1. Data Encryption Standard (DES),
   2. RC2, RC4, RC5, RC6,
   3. International Data Encyrption Standard (IDEA),
   4. Advanced Encryption Standard (AES),
   5. One Time Pad (OTP)

2. Algoritma Asimetri

   Algoritma asimetri sering juga disebut dengan algoritma kunci public, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian, yaitu :

1. Kunci umum *(public key);* Kunci yang boleh semua orang tahu (dipublikasikan).
2. Kunci rahasia *(private key)*; Kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).

3. Fungsi Hash

Fungsi Hash sering disebut fungsi Hash satu arah (one-away function), message digest, fingerprint, fungsi kompresi dan message authentication code (MAD). Merupakan suatu fungsi matematika yang mengambil masukan panjang variable dan mengubahnya kedalam urutan biner dengan panjang yang tetap.

2.7 Pengertian *Stream Cipher*

Kriptografi dibagi dalam dua bagian, yaitu *cipher modern* dan klasikal. Mode klasikal biasanya menggunakan mode karakter, sehingga kemampuan dari mode ini terbatas karena keterbatasan jumlah karakter.

Stream Cipher ditemukan oleh *Gilbert Vernam* pada tahun 1917, meskipun algoritma saat itu belum bisa disebut dengan demikian pada saat itu. Ia membangun sebuah mesin elektromagnetik yang mana secara otomatis mengenkripsi komunikasi dari mesin ketik teletip. *Plaintext* akan masuk kedalam mesin menjadi sebuah kertas *tape*, dan *key stream* sebagai *tape* kedua. Proses tersebut adalah pertama kalinya antara enkripsi dan transmisi dikerjakan secara otomatis dalam satu mesin.

Perkembangan algoritma kriptografi modern berbasis bit didorong oleh penggunaan komputer digital yang merepresentasikan data dalam bentuk biner. Algoritma kriptografi yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua kategori yaitu *block cipher* dan *stream cipher*.

Stream Cipher (aliran cipher) merupakan suatu cipher yang berasal dari hasil XOR . Setiap bit plaintext dengan dengan setiap bit kunci. Kunci merupakan kunci utama (kunci induk) yang digunakan untuk membangkitkan kunci acak semu yang dibangkitkan dengan *Pseudo-Random Squence Generator* yang merupakan suatu nilai yang nampak seperti di acak, tetapi sesungguhnya nilai tersebut merupakan suatu urutan. Secara khusus urutan dari nilai yang dihasilkan oleh RNG *(random number generator),* computational mekanisme deterministic atau FSM *(finite state machine)* merupakan kebalikan dari really random (Doni Ariyus, 2005).

Random Number Generato (RNG) secara umum adalah *Pseudorandom*. Yang memberikan *initial state* atau *seed* (nilai yang diinputkan kedalam *state*), seluruh urutan tersebut ditentukan secara keseluruhan, tetapi meskipun demikian banyaknya karakteristik yang ditampilkan dari suatu urutan yang acak tersebut. *Pseudorandomness* menghasilkan yang sama secara berulang-ulang pada penempatan yang berbeda. Kemudian kunci acak semu tersebut diberikan operasi XOR dengan plaintext untuk mendapatkan ciphertext.

Stream Cipher rawan terhadap serangan pembalikan bit. Jika penyerang mengetahui bahwa stream cipher yang digunakan, penyerang mencoba untuk mengedintifikasi posisi bit yang telah dirubah dan mengembalikannya kebentuk

aslinya, mereka terlebih dahulu mencari polda dari perubahan bit tersebut untuk mengedintifikasi bentuk asli dari bit tersebut (Doni Ariyus, 2005).

2.8 Algoritma RC4

Algoritma *RC4* merupakan salah satu jenis *stream cipher,* yaitu memproses unit atau input data, pesan atau informasi pada suatu saat. Dengan cara ini enkripsi dan deskripsi dapat dapat dilakukan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah input data, pesan atau informasi tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk mengenkrip, Contoh *Stream cipher* adalah *RC4, Seal, A5. Oryc,* dan lain-lain. Tipe lainnya adalah *block cipher* yang memproses sekaligus sejumlah data tertentu, biasanya 64 bit atau 128 bit blok, contohnya : *Blowfish, DES, Gost, RC5* dan lain-lain (Pandiangan dan Sijabat, 2016).

RC4 merupakan enkripsi *stream* simetrik *proprietary* yang dibuat oleh *RSA Data Security Inc (RSADSI).* Penyebarannya diawali dari sebuah source code yang diyakini sebagai RC4 dan dipublikasikan secara *'anonymously'* pada tahun 1994, Algoritma yang dipublikasikan ini sangat identik dengan implementasi *RC4* pada produk resmi. *RC4* digunakan secara luas pada beberapa aplikasi dan pada umumnnya dikatakan sangat aman. Sampai saat ini diketahui tidak ada yang dapat memecahkan/membongkarnya, RC4 tidak dipatenkan oleh RSADSI, hanya saja tidak diperdagangkan secara bebas (Slamet Maryono, 2012).

Algoritma RC4 cukup mudah umtuk dijelaskan. RC4 mempunyai sebuah S-Box,S0,S1,….S255, yang berisi permutasi dari bilangan 0 sampai 255, dan

permutasi merupakan fungsi dari kunci dengan panjang yang variable. Terdapat dua indeks yaitu i dan j, yang diinsialisasi dengan bilangan nol. Untuk menghasilkan random byte langkahnya sebagai berikut (Slamet Maryono, 2012).

i=(i+1)mod256

j=(j+S1)mod256

swap Si dan Sj

t=(Si+Sj)mod256

K=St

Algoritma *RC4* yang mengenkripsi antara kombinasi plainteks dengan menggunakan *bit-wise Xor (Exclusive-or). RC4* menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan *XOR* dengan *plaintext* untuk menghasilkan *ciphertext*. Masing - masing elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsinya dilakukan dengan cara yang sama (karena Xor merupakan fungsi simetrik).

Untuk menghasilkan *keystream*, *ciphe*r menggunakan *state internal* yang meliputi dua bagian :

1. Tahap *key scheduling algoritihm* (KSA) dimana diberi nilai awal berdasarkan kunci enkripsi. State yang diberi nilai awal berupa array yang merepresentasikan suatu permutasi dengan 256 elemen, jadi hasil dari algoritma KSA adalah permutasi awal. Array yang mempunyai 256 elemen ini (dengan indeks 0 sampai dengan 255) dinamakan S. Berikut adalah algoritma KSA dalam bentuk *pseudo-*

code dimana key adalah kunci enkripsi dan *keylength* adalah besar kunci enkripsi dalam *bytes* (untuk kunci 128 bit, *keylength* = 16).

Proses penjadwalan kunci (*key Schedulling algorithm*) dilakukan dengan tujuan membangkitkan kunci yang acak sejumlah 256 buah kunci. Penjadwalan kunci melibatkan dua tabel array yaitu *array* S dan *array* T. Proses pengacakan dilakukan dengan menukarkan nilai-nilai *array* S yang sebelumnya dikalkulasikan dengan nilai-nilai array T. Adapun *pseudocode* untuk melakukan pembentukan *array* S dan T adalah:

```
for(i=0;i<=255;i++){
    S-Box[i]=i
    T[i] = kunci [ I mod panjang_kunci]
}
```

Pseudocode untuk melakukan permutasi *array* S adalah :

```
j=0
for(i=0;i<=255;i++){
    j=(j+S-Box[i]+T[i])mod 256
    Swap(S-Box[i],S[j])
    j=j
}
```

2. *Pseudo Random Generation Algorithm* (PRGA). Proses *pseudo random generation* merupakan proses yang dilakukan untuk membangkitkan kunci sebanyak elemen *plaintext* yang akan

dienkripsi. Proses ini melibatkan nilai-nilai pada tabel *array* S yang telah dipermutasi (diacak). Kunci- kunci ini lah nantinya yang akan di XOR-kan dengan *plaintext*. Adapun *pseudo code* untuk melakukan PRGA adalah :

```
i = 0; j = i
for (i = 0 ; i <= jlh_karakter_plaintext; i++){ i = (i + 1) mod 256
j = (j + S-Box[i]) mod
256 Swap( S-Box[i], S-
Box[j] )
t = (S-Box[i] + S-Box[j]) mod
256 Kunci[i] = S-Box[t]
}
```

Kunci enkripsi didapat dari sebuat 256 bit *state-array (KSA)* yang diinisialisasi dengan sebuah *key* tersendiri dengan panjang 1-256 bit. Setelah itu, *state-array* tersebut akan diacak kembali dan diproses untuk menghasilkan sebuah kunci enkripsi yang akan di-XOR-kan dengan plainteks ataupun cipherteks. Secara umum, algoritma RC4 terbagi menjadi dua, inisalisasi *state- array* dan penghasilan kunci enkripsi serta pengenkripsiannya. Adapun algoritma dari Enkripsi RC-4 adalah sebagai berikut ini :

Langkah 1:

Inisialisasi S-Box (Array S) Jum = Len(Kunci)

i = 0

j = 0

For y=1 to Jum

j = (j + S[i] + K [i mod jum]) mod jum

Swap (S[i],S[j])

Next y

Langkah 2:

Lakukan Pengacakan S-Box Jum = Len(Kunci)

i = 0

j = 0

For y=1 to Jum

i = (i + 1) mod Jum

j = (j + S[i]) mod Jum

swap (S[i],S[j])

Key(y) = S[(S[i]+S[j]) mod jum]

Next y

Langkah 3:

Lakukan Enkripsi

Jum = Panjang(PlainText) i = 0

j = 0

For y=1 to Jum

C(y) = Biner(P[y]) XOR Biner(Key[y]) Next y

Dan untuk melakukan Dekripsi maka dilakukan langkah sebagai berikut ini :

Langkah 1:

Inisialisasi S-Box (Array S) Jum = Len(Kunci)

i = 0

j = 0

For y=1 to Jum

j = (j + S[i] + K [i mod jum]) mod jum Swap (S[i],S[j])

Next y

Langkah 2:

Lakukan Pengacakan S-Box Jum = Len(Kunci)

i = 0

j = 0

For y=1 to Jum

i = (i + 1) mod Jum

j = (j + S[i]) mod Jum swap (S[i],S[j])

Key(y) = S[(S[i]+S[j]) mod jum]

Next y

Langkah 3:

Lakukan Enkripsi

Jum = Panjang(PlainText) i = 0

j = 0

For y=1 to Jum

P(y) = Biner(C[y]) XOR Biner(Key[y]) Next y

2.9 Pengertian *SMS* (*Short Message Service*)

Sejarah *SMS* muncul pada Desember 1992, *SMS* ialah teknologi yang mampu mengirim dan menerima pesan antara telepon seluler. *SMS* pertama dikirimkan oleh Neil Papwort kepada Richard Jarvis melalui komputer ke sebuah telepon selular dalam jaringan *GSM* milik operator seluler *Vodafone* di Inggris.

SMS (Short Message Service) adalah teknologi yang dapat mengirimkan pesan antara telepon seluler dengan yang lainnya. Ada beberapa provide seluler yang terdapat di Indonesia antara lain Telkomsel, XL, Indosat, AXIS, 3 dan lainnya (Frangky Rawung, 2017).

2.9.1 Cara Kerja *SMS*

Mekanisme cara kerja sistem SMS adalah melakukan *short message* satu terminal pelanggan ke terminal lainnya. Hal ini dapat dilakukan karena adanya entitas dalam sistem *SMS* yang bernama *Short Message Service Center (SMSC)*, disebut juga *Message Center (MC)*. *SMSC* merupakan sebuah perangkat yang melakukan penerimaan dan pencarian rute tujuan akhir dari *SMS* (Diana dan Zebua, 2018).

1. *Short Message Service Center (SMSC)*

   Menurut Gunawan, Pada saat pengiriman *SMS* dari telepon seluler, *SMS* itu tidak langsung dikirim pada telpon seluler tujuan, akan tetapi dikirim terlebih dahulu ke *SMS Center*, lalu *SMS* tersebut diteruskan pada telepon tujuan.

2. *Home Location Register (HLR)*

   Sebelum *SMS Center* mengirimkan pesan yang anda buat, *SMS Center* terlebih dahulu mengirimkan *SMS request* ke *HLR* yaitu perangkat yang berisikan data detail untuk setiap *subscribe*r melalui *STP (Signal Transfer Point)* untuk menemukan pelanggan tujuan anda.

3. *Period Validity*

   SMS akan tersimpan di *SMS Center* jika nomor tujuan anda tidak aktif atau diluar jaringan sampai *period validity* terpenuhi. *Period validity* yaitu tenggang waktu yang diberikan sampai pesan diterima oleh si penerima (nomor tujuan). Setelah *period validity* penuh dan nomor tujuan masih belum aktif, maka *SMS Center* akan mengirim pesan "*SMS* gagal terkirim" jika terkirim maka *SMS Center* mengirimkan "*SMS* berhasil terkirim".

2.9.2 Pengertian *SMS Gateway*

SMS Gateway memungkinkan kita untuk mengirim dan menerima SMS secara progmmatically (dari aplikasi) dan menjadikan smartphone Android kita sebagai perangkatnya. Seperti perangkat tambahan modem misalnya. Kita harus menancapkan modem tersebut pada sebuah PC atau laptop selama 24 jam penuh (atau selama SMS digunakan). Ini seperti kita membangun server mini. Kelebihannya, kita bisa mengontrol SMS masuk mauopun keluar sesuka hati.

SMS Gateway adalah proses pengiriman pesan dengan format yang telah ditentukan dengan mendapatkan balasan pesan secara langsung yang dikirmkan ke dalam sms center (Frangky Rawung, 2017).

Adapun jenis-jenis SMS Gateway antara lain:

1. SMS Auto Informasi
2. SMS Jadwal
3. SMS Shutdown PC
4. SMS Polling dan lainnya.

SMS Gateway sangat memberikan manfaat bagi kegiatan kita sehari-hari antara lain:

1. SMS Gateway sangat mudah untuk digunakan untuk sekolah, instansi, perusahaan, pertokoan dll.
2. Harga murah dan cepat.
3. Kita dapat menggunkan SMS Gateway kapan dan dimanapun.

2.10 Pengertian *XAMPP*

XAMPP adalah sebuah aplikasi *web server* instan dan lengkap dikarenakan segala yang ada butuh untuk membuat sebuah situs *web* dengan *content management system* bisa dicoba dalam aplikasi ini. *XAMPP* merupakan sebuah paket *insteller AMP (Apache, Mysql, PHP)* yang sangat mudah diaplikasikan dalam komputer anda menggunakan bahasa *server* dan *database server* tersebut (Frangky Rawung, 2017).

2.11 Pengertian *PHP*

PHP merupakan bahasa pemrograman yang banyak digunakan untuk membuat web yang dinamis atau yang memiliki kepanjangan *Hypertext preprocessor*. PHP adalah software yang diperoleh secara gratis karena bersifat *open source* dan dapat digunakan berbagai jenis platform sistem operasi. *PHP* merupakan bahasa pemrograman yang dapat disisipkan kedalam skrip *HTML* untuk membuat web dinamis dengan cepat. Untuk menjalankan bahasa pemrograman *PHP*, kita memerlukan web server untuk dapat menjalankannya (Frangky Rawung, 2017).

Untuk dapat menggunakan PHP, ada beberapa aplikasi yang harus diperlukan seperti berikut:

1. Web Server
2. Database (MySQL. Oracle, dll)
3. Web Editor (Notepad++, Dreamweaver, VSCode, dll)
4. Web Browser (Mozilla, Internet Explorer, Opera, dll).

2.12 Pengertian *HTML*

Hypertext MarkUp Language (HTML) adalah bahasa pemrograman yang dipakai untuk menampilkan informasi pada halaman web menurut (Sitorus, 2012).

Sedangkan *HTML* adalah aplikasi *SGML* yaitu untuk mengedintifikasi tipe dokumen terstruktur dan menetapkan bahasa untuk mempresentasikan tipe dokumen (Indrajani, 2011).

2.13 Pengertian *Mysql*

MySql atau dibaca My Sekuel dalah suatu RDBMS *(Relational Data-Base Management System)* yaitu aplikasi sistem yang menjalankan fungsi pengolahan data (Sibero, 2014).

Mysql adalah perangkat lunak (*software)* manajemen database *open source* untuk digunakan sebagai menambahkan, mengupdate, menghapus dan menampilkan data (Frangky Rawung, 2017).

Tabel 2.1 Perintah dan keterangan dalam MySql

Perintah	Keterangan
Show database	Perintah ini untuk menampilkan atau melihat daftar database yang sudah ada (sudah dibuat)
Use	Perintah ini digunakan untuk masuk atau mengakses database yang sudah ada.
Show tables	Perintah ini digunakan untuk melihat atau menampilkan semua yang ada di dalam database aktif
Desc/describel	Perintah ini digunakan untuk melihat struktur table.
Quit	Perintah ini dinggunakan untuk keluar dari mysql server

Sumber : Bunafit, 2013

2.14 Pengertian *UML*

UML (Unified Modeling Language) terdiri dari 13 macam diagram yang dikelompokan dalam 3 katagori. Berikut penjelasan singkat dari pembagian katagori tersebut (Rosa A.S dan M. Shalahudin, 2014).

1. *Structure* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodalkan. *Structure* diagram *terdiri dari class* diagram, *object* diagram, *component* diagram, *composite structure* diagram, *package* diagram dan *deployment* diagram. *Class* Diagram menggambarkan struktur sistem dari segi pendefisinian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.
2. *Behavior* diagram yaitu kumpulan yang digunakan untuk menggambarkan keluaran sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behavior* diagram terdiri dari *use case* diagram, *activity* diagram, *state machine system. Use case* diagram berfungsi untuk menggambarkan kegiatan aktor atau pengguna aplikasi, sedangkan *Activity* diagram menggambarkan proses alur kerja aktivitas, diagram ini sangat mirip dengan *flowchart* karena dapat memodelkan proses logika dan alur kerja. Terdapat beberapa menu yang ditampilkan ataupun proses yang terjadi setelah pengguna menjalankan aplikasi.

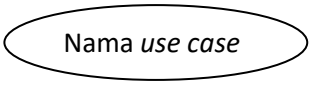
3. *Interaction* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun intraksi antara *sub*sistem pada suatu sistem. Interaction diagram terdiri dari *sequence* diagram, *communication* diagram, *timing* diagram, *interaction overview* diagram. *Sequence Diagram* menggambarkan perilaku objek *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

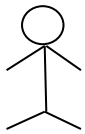
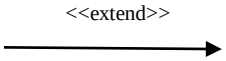
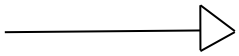
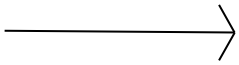
2.15 *Use Case* Diagram

Use case merupakan suatu diagram yang berisi use case, actor, serta relationship/hubungan di dalamnya, Adalah titik awal memahami atau menganalisis kebutuhan sistem dan dapat menggambarkan dengan detail bagaimana suatu sistem memproses atau melakukan suatu perancangan sistem (Nugroho Adi, 2005).

Berikut adalah simbol-simbol yang ada pada *use case* diagram :

Tabel 2.2 *Use Case* diagram

No	Simbol	Diskripsi
1	*Use case* 	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *user case*

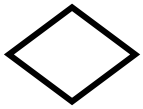
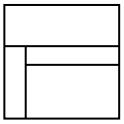
2	*Aktor* 	Orang, proses,atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	Asosiasi	Kombinasi antara aktor dan *use case* atau *use case* memiliki intraksi dengan aktor.
4	Ekstensi 	Relasi *use case* tambahkan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu mirip dengan prisip *inheritance* pada pemrogramannya.
5	Generalisasi 	Hubungan generalisasi dan sepesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Mengunakan 	Relasi use case tambahkan kesebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan funsi atau sebagian syarat dijalankan use case.

Sumber: Rosa A.S dan M.Shalahudin, 2014

2.16 *Activity* Diagram

Diagram aktivitas atau *activity* diagram yang menunjukkan aliran kerja atau aktifitas dari sebuah sistem atau proses bisnis atau menuyang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram *activity* (Rose dan M.Shalahudin, 2014).

Tabel 2.3 simbol-simbol *activity* diagram

No	Simbol	Keterangan
1	Status awal 	Status awal aktifitas sistem.
2	Aktivitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja.
3	Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas status akhir.
6	*Swimlane* 	*Swimlane* memisahkan organisasi bisnis yang bertanggung jawab

		terhadap aktivitas yang terjadi.

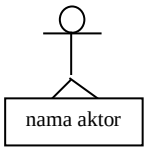
Sumber: Rosa A.S dan M.Shalahudin, 2014

2.17 *Sequence* diagram

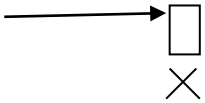
Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan massage yang dikirimkan dan ditrima antar objek (Rose dan M.Shalahudin, 2014).

Berikut adalah simbol-simbol yang ada pada *sequence* diagram :

Tabel 2.4 *Sequence* diagram

No	Simbol	Keterangan
1	Aktor atau  tanpa waktu aktif	Orang, proses,atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
2	Garis hidup 	Menyatakan kehidupan suatu objek

3	Objek Nama objek: nama kelas	Menyatakan objek yang berintaksi pesan.
4	Waktu aktif 	Menyatakan objek dalam keaddan aktif dan berintraksi, semuanya yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5	Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah menyatakan pada objek yang dibuat.
6	Pesan tipe *call* 1: nama metode () 	Menyatakan suatu objek memanggil oprasi/metode yang ada pada objek lainnya atau dirinya sendiri.
7	Pesan tipe *send* 1 : masukan 	Menyatakan bahwa suatu objek mengirim data/masukan/informasi kedalam objek lain

8	pesan tipe return 1 : keluaran 	Mnyatakan bahwa suatu objek yang telah menjalankan suatu oprasi atau metode penghasilan suatu pengembalian keobjek tertentu
9	Pesan tipe *destrory* 	Menyataka suatu objek mengakhiri hidup objek yang lainnya, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* maka ada *destroy*

Sumber: Rosa A.S dan M.Shalahudin, 2014

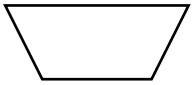
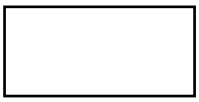
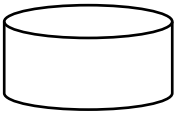
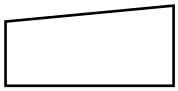
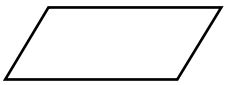
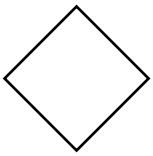
2.18 Pengertian *Flowchart*

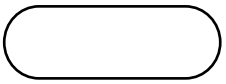
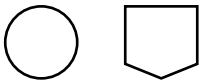
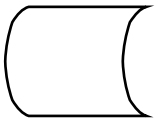
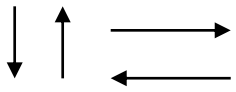
Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program yang mempunyai arus yang menggambarkan langkah-langkah suatu masalah (Jogiyanto, 2005).

Flowchat digunakan untuk mempermudah penyusunan program. Dengan menggunakan *Flowchat*, logika pemrograman lebih mudah dipahami dan dianalisis, sehingga anda dapat menentukan kode-kode pemrograman yang sesuai

Menjelaskan simbol-simbol dalam *Flowchat* adalah sebagai berikut (Jogiyanto, 2005).

Tabel 2.5 Simbol simbol dalam *Flowchart*

Simbol	Arti
	Simbol dokumen I/O manual, mekanik atau komputer.
	Simbol kegiatan menual.
	Simbol proses dari program komputer.
	Simbol harddisk.
	Simbol keyboard.
	Input/Output.
	Keputusan.
	Proses terdefinisi, menunjukkan suaru oprasi yang rinciannya ada di tempat lain.

	Persiapan, untuk memberi nilai awal suatu besaran
	Terminal, untuk menunjukkan awal dan akhir suatu proses.
	Terminal yang mewakili simbol tertentu untuk untuk digunakan pada aliran yang lain pada halaman yang sama.
	Simbol disket
	Simbol aliran data, menggambarkan aliran data yang berupa masukkan untuk sistem atau hasil dari proses sistem.

Sumber : Jogiyanto, 2005

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Skripsi ini dikerjakan dengan menggunakan metodologi penelitian sebagai berikut :

1. Mempelajari bermacam sumber literatur. Yaitu dari beberapa sumber buku,*e-book* dan *internet* khususnya yang yang berhubungan dengan *Stream Cipher*.
2. Konsultasi dengan dosen pembimbing dan pihak-pihak lain yang bisa membantu.
3. Pencarian data melalui referensi skripsi alumni yang terdapat di perpustakaan UNPAB.
4. Mencoba menginputkan berbagai jenis *file* dengan berbagai ukuran yang beragam agar diketahui seberapa efektifkah program yang telah dibuat.

3.2 Analisis Sistem

Analisa sistem merupakan fase awal untuk pengembangan sistem. Tahap ini mempelajari sebuah sistem dengan menguraikan komponen-komponen di dalamnnya dan bagaimana seluruh sistem berinteraksi sehingga sistem yang dibangun dapat mencapai tujuan yang diharapkan.

3.3 Analisis Masalah

Kebutuhan manusia akan informasi dan komunikasi seakan menjadi kebutuhan yang tidak dapat dipisahkan dalam kehidupan sehari-hari. Pemanfaatan SMS pada kegunaanya memungkinkan setiap pengguna dalam mengirim dan menerima informasi dengan cepat. Pengamanan juga menjadi aspek penting yang perlu diperhatikan dalam penggunaan SMS.

Salah satu dari teknik mengamankan suatu pesan yaitu dengan cara teknik kriptografi yang menyandikan pesan sebagai kode aneh yang membuat penyusup yang melihatnya menjadi penasaran dan berusaha untuk mengetahui kode itu. Untuk menambah tingkat keamanan suatu pesan, maka dilakukanlah teknik kriptografi dengan *stream cipher.* Algoritma kriptografi yang digunakan disini adalah algoritma *RC4.*

3.4 Teknik Pemecahan Masalah

Teknik pemecahan masalah tentang perancangan aplikasi keaman sms yang dibuat memiliki poin yaitu sebagai berikut:

1. Tahap pertama analisa terhadap perancangan yang akan dibangun terutama tentang keamanan *database* sms.
2. Tahap kedua yaitu aplikasi dalam persiapan menentukan perangkat yang dibutuhkan dalam membangun aplikasi seperti perangkat keras *(hardware)* maupun lunak *(software).*

3. Kemudian dilakukan perancangan sistem yang nantinya akan di implementasikan pada aplikasi yang akan dibangun.
4. Terakhir adalah proses uji coba yang akan dilakukan inputan, proses ataupun output aplikasi, apakah sudah sesuai dengan perancangan yang telah direncakan sebelumnya.

3.5 Analisis Proses Penyelesaian

Ukuran sandi *RC4* sangat berpengaruh terhadap keamanan sistem Sandi *RC4*. *RC4* telah dibuktikan bisa mengamankan untuk ukuran kunci yang kecil. Rekomendasi penggunaan sistem sandi *RC4* agar memiliki keamanan yang kuat adalah:

1. Ukuran kunci sama atau lebih besar daripada 256 *bit*
2. Setiap sesi baru membangkitkan kunci yang baru (dengan pembangkitkan kunci yang baru menghingdari penyerang untuk melakukan analisis sandi difernsial pada sistem sandi).

Berikut adalah implementasi algoritma RC4 dengan mode 4 *byte* (untuk lebih menyederhanakan dalam perhitungan manual) serta untuk kebutuhan sistem yang sangat terbatas. S-Box dengan panjang 4 byte, dengan S[0]=0, S[1]=1, S[2]=2 dan S[3]=3 sehingga array S menjadi 0 1 2 3.

3.6 Implementasi

Terdiri Inisialisasi 13 byte kunci array K. Dan mencoba untuk mengenkripsikan kata MUHAMMADRIZKY dengan kunci ASCII (77)(85)(72)(65)(77)(77)(65)(68)(82)(73)(90)(75)(89)

Array S	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Array K	**77**	**85**	**72**	**65**	**77**	**77**	**65**	**68**	**82**	**73**	**90**	**75**	**89**

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA agar tercipta state-array yang acak. Penjelasan iterasi lebih lanjut dapat dijelaskan sebagai:

Iterasi 1:

i = 0

j = (0+S[0]+K(0mod13)) mod13

= (0+0+77)mod13= 12

Swap (S[0],S[12])

Array S	**12**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**0**

Iterasi 2

i = 1

j = (12+S[1]+K(2mod13)) mod13

= (12+1+85) mod13 = 98 mod13 = 7

Swap (S[1],S[7])

Array S	**12**	**7**	**2**	**3**	**4**	**5**	**6**	**1**	**8**	**9**	**10**	**11**	**0**

Iterasi 3

i = 2

j = (7+S[2]+K(2mod13)) mod13

= (7+2+72)mod13= 81 mod13 = 3

Swap (S[2],S[3])

Array S	**12**	**7**	**3**	**2**	**4**	**5**	**6**	**1**	**8**	**9**	**10**	**11**	**0**

Iterasi 4

i = 3

j = (3+S[3]+K(3mod13)) mod13

= (3+2+65)mod13= 70 mod13 =5

Swap (S[3],S[5])

Array S	**12**	**7**	**3**	**5**	**4**	**2**	**6**	**1**	**8**	**9**	**10**	**11**	**0**

Iterasi 5

i = 4

j = (5+S[4]+K(4mod13)) mod13

= (5+4+77)mod13 = 86 mod13 = 8

Swap (S[4],S[8])

Array S	**12**	**7**	**3**	**5**	**8**	**2**	**6**	**1**	**4**	**9**	**10**	**11**	**0**

Iterasi 6

i = 5

j = (8+S[5]+K(5mod13)) mod13

= (8+2+77)mod13 = 87 mod13 = 9

Swap (S[5],S[9])

Array S	**12**	**7**	**3**	**5**	**8**	**9**	**6**	**1**	**4**	**2**	**10**	**11**	**0**

Iterasi 7

i = 6

j = (9+S[6]+K(6mod13)) mod13

= (9+6+65)mod13 = 80 mod13 = 2

Swap (S[6],S[2])

Array S	**12**	**7**	**6**	**5**	**8**	**9**	**3**	**1**	**4**	**2**	**10**	**11**	**0**

Iterasi 8

i = 7

j = (2+S[7]+K(7mod13)) mod13

= (2+1+68)mod13 = 71 mod 13 = 6

Swap (S[7],S[6])

Array S	**12**	**7**	**6**	**5**	**8**	**9**	**1**	**3**	**4**	**2**	**10**	**11**	**0**

Iterasi 9

i = 8

j = (6+S[8]+K(8mod13)) mod13

= (6+4+82)mod13 = 92 mod13 = 1

Swap (S[8],S[1])

Array S	**12**	**4**	**6**	**5**	**8**	**9**	**1**	**3**	**7**	**2**	**10**	**11**	**0**

Iterasi 10

i = 9

j = (1+S[9]+K(9mod13)) mod13

= (1+2+73)mod13 = 76 mod13 = 11

Swap (S[9],S[11])

Array S	**12**	**4**	**6**	**5**	**8**	**9**	**1**	**3**	**7**	**11**	**10**	**2**	**0**

Iterasi 11

i = 10

j = (11+S[10]+K(10mod13)) mod13

= (11+10+90)mod13 = 111 mod13 = 7

Swap (S[10],S[7])

Array S	**12**	**4**	**6**	**5**	**8**	**9**	**1**	**10**	**7**	**11**	**3**	**2**	**0**

Iterasi 12

i = 11

j = (7+S[11]+K(11mod13)) mod13

= (7+2+75)mod13 = 84 mod13 = 6

Swap (S[11],S[6])

Array S	**12**	**4**	**6**	**5**	**8**	**9**	**2**	**10**	**7**	**11**	**3**	**1**	**0**

Iterasi 13

i = 12

j = (6+S[12]+K(12mod13)) mod13

= (6+0+89)mod13= 95 mod13 = 4

Swap (S[12],S[4])

Array S	**12**	**4**	**6**	**5**	**0**	**9**	**2**	**10**	**7**	**11**	**3**	**1**	**8**

Setelah melakukan KSA, akan dilakukan PRGA. PRGA akan dilakukan sebanyak 13 kali dikarenakan plainteks yang akan dienkripsi berjumlah 13 karakter. Hal ini disebakan karena dibutuhkan 1 kunci dan 1 kali pengoperasian XOR untuk tiap tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

Array S	**12**	**4**	**6**	**5**	**0**	**9**	**2**	**10**	**7**	**11**	**3**	**1**	**8**

Inisialisasi

i = 0

j = 0

iterasi 1

i = (0+1)mod13 = 1

j = (0+S[1])mod13

= (0+4)mod13 = 4

Swap (S[1],S[4])

12	**0**	**6**	**5**	**4**	**9**	**2**	**10**	**7**	**11**	**3**	**1**	**8**

K1 = S([1]+S[4]mod13)

= S[4mod13]

=S[4]

= 4

K1 = 00110100

iterasi 2

i = (1+1)mod13 = 2

j = (4+S[2])mod13

= (4+6)mod13 = 10

Swap (S[2],S[10])

12	**0**	**3**	**5**	**4**	**9**	**2**	**10**	**7**	**11**	**6**	**1**	**8**

K2 = S(S[2]+S[10]mod13)

= S(3+6 mod13)

= S[9]

= 11

K2 = 00110001 00110001

iterasi 3

i = (2+1)mod13 = 3

j = (10+S[3])mod13

= (10+5)mod13 = 2

Swap (S[3],S[2])

12	**0**	**5**	**3**	**4**	**9**	**2**	**10**	**7**	**11**	**6**	**1**	**8**

K3 = S(S[3]+S[2]mod13)

= S(3+5)mod13

= S(8 mod13)

= S[8]

= 7

K2 = 00110111

iterasi 4

i = (3+1)mod13 = 4

j = (2+S[4])mod13

= (2+4)mod13 = 6

Swap (S[4],S[6])

12	**0**	**5**	**3**	**2**	**9**	**4**	**10**	**7**	**11**	**6**	**1**	**8**

K4 = S(S[4]+S[6]mod13)

= S(2+4)mod13

= S(6 mod13)

= S[6]

= 4

K4 = 00110100

iterasi 5

i = (4+1)mod13 = 5

j = (6+S[5])mod13

= (6+9)mod13 = 2

Swap (S[5],S[2])

12	**0**	**9**	**3**	**2**	**5**	**4**	**10**	**7**	**11**	**6**	**1**	**8**

K5 = S(S[5]+S[2]mod13)

= S(5+9)mod13

= S[1]

= 0

K5 = 00110000

iterasi 6

i = (5+1)mod13 = 6

j = (2+S[6])mod13

= (2+4)mod13 = 6

Swap (S[6],S[6])

12	**0**	**9**	**3**	**2**	**5**	**4**	**10**	**7**	**11**	**6**	**1**	**8**

K6 = S(S[6]+S[6]mod13)

= S(4+4)mod13

= S[8]

= 7

K6 = 00110111

iterasi 7

i = (6+1)mod13 = 7

j = (6+S[7])mod13

= (6+10)mod13 = 3

Swap (S[7],S[3])

12	**0**	**9**	**10**	**2**	**5**	**4**	**3**	**7**	**11**	**6**	**1**	**8**

K7 = S(S[7]+S[3]mod13)

= S(3+10)mod13

= S[0]

= 12

K7 = 00110001 00110010

iterasi 8

i = (7+1)mod13 = 8

j = (3+S[8])mod13

= (3+7)mod13 = 10

Swap (S[8],S[10])

12	**0**	**9**	**10**	**2**	**5**	**4**	**3**	**6**	**11**	**7**	**1**	**8**

K8 = S(S[8]+S[10]mod13)

= S(6+7)mod13

= S[0]

= 12

K8 = 00110001 00110010

iterasi 9

i = (8+1)mod13 = 9

j = (10+S[9])mod13

= (10+11)mod13 = 8

Swap (S[9],S[8])

12	**0**	**9**	**10**	**2**	**5**	**4**	**3**	**11**	**6**	**7**	**1**	**8**

K9 = S(S[9]+S[8]mod13)

= S(6+11)mod13

= S[4]

= 2

K9 = 00110010

iterasi 10

i = (9+1)mod13 = 10

j = (8+S[10])mod13

= (8+7)mod13 = 2

Swap (S[10],S[2])

12	**0**	**7**	**10**	**2**	**5**	**4**	**3**	**11**	**6**	**9**	**1**	**8**

K10 = S(S[10]+S[2]mod13)

= S(9+7)mod13

= S[3]

= 10

K10 = 00110001 00110000

iterasi 11

i = (10+1)mod13 = 11

j = (2+S[11])mod13

= (2+1)mod13 = 3

Swap (S[11],S[3])

12	**0**	**7**	**9**	**2**	**5**	**4**	**3**	**11**	**6**	**10**	**1**	**8**

K11 = S(S[11]+S[3]mod13)

= S(1+9)mod13

= S[10]

= 10

K11 = 00110001 00110000

iterasi 12

i = (11+1)mod13 = 12

j = (3+S[12])mod13

= (3+8)mod13 = 11

Swap (S[12],S[11])

12	**0**	**7**	**9**	**2**	**5**	**4**	**3**	**11**	**6**	**10**	**8**	**1**

K12 = S(S[12]+S[11]mod13)

= S(8+1)mod13

= S[9]

= 6

K12 = 00110110

iterasi 13

i = (12+1)mod13 = 13

j = (11+S[0])mod13

= (11+12)mod13 = 10

Swap (S[0],S[10])

10	**0**	**7**	**9**	**2**	**5**	**4**	**3**	**11**	**6**	**12**	**8**	**1**

K13 = S(S[0]+S[10]mod13)

= S(10+12)mod13

= S[9]

= 6

K13 = 00110110

Setelah menemukan kunci untuk tiap karakter, makadilakukan operasi XOR antara karakter pada plaintext dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada plaintks yang digunakan.

Huruf Kode ASCII	(Binary 8 bit)
M	01001101
U	01010101
H	01001000
A	01000001
M	01001101
M	01001101
A	01000001
D	01000100

R	01010010
I	01001001
Z	01011010
K	01001011
Y	01011001

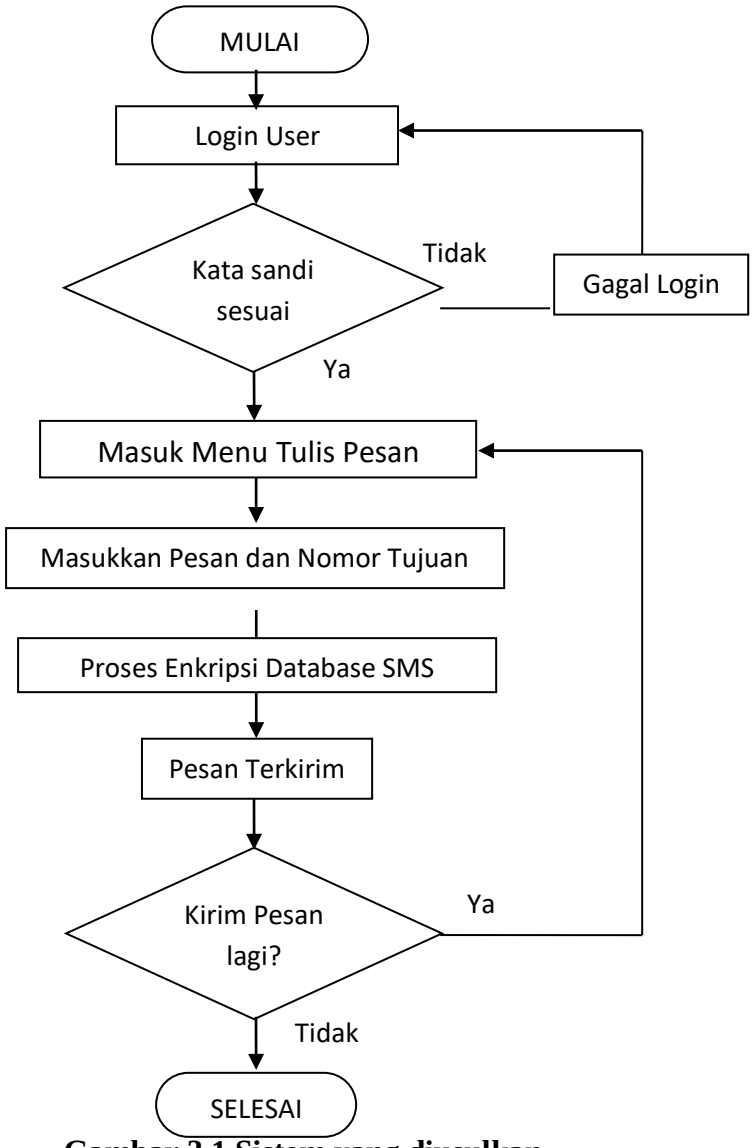
Berikut adalah proses peng-XORan dari plainteks dengan key yang telah didapat:

MUHAMMAD RIZKY	(01001101)(01010101)(01001000) (01000001)(01001101)(01001101) (01000001)(01000100)(01010010) (01001001)(01011010)(01001011) (01011001)
Key	(00110100)M (00110001 00110001)U (00110111)H (00110100)A (00110000)M (00110111)M (00110001 00110010)D (00110010)R (00110001)I (00110001 00110000)Z (00110110)K (00110110)Y
Cipherteks	(01111001)y (01100101 01100101)ee (01111111)# (01110101)u (01111101) } (01111011) { (01110000 01110010) pr (01110101 01110110) uv

	(01100000) ‘ (01111000 01111001) xö (01101011 01101010) k5 (01111101) } (01101111) o

3.7 Analisis Sistem Yang Diusulkan

Ada pun sistem yang diusulkan sebagai berikut:



Gambar 3.1 Sistem yang diusulkan

3.8 Analisis Perangkat Lunak (*software*)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang dirancang untuk memudahkan dalam membangun dan menjalankan sistem.

Adapun perangkat lunak yang digunakan adalah sebagai berikut:

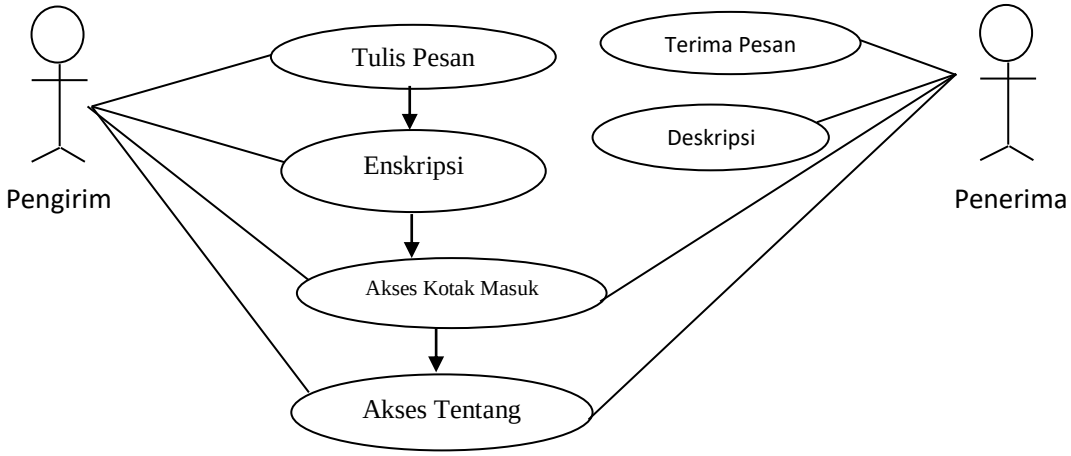
1. *Microsoft Windows 7, Windows xp* sebagai sistem Operasi
2. *Mozila Firefox* version 3.5 sebagai browser
3. Modem untuk koneksi internet.

3.9 Perancangan Sistem

Tujuan perancangan untuk mengetahui gambaran alur yang akan di bangun serta proses yang ada di dalam nya.

3.9.1 *Use Case*

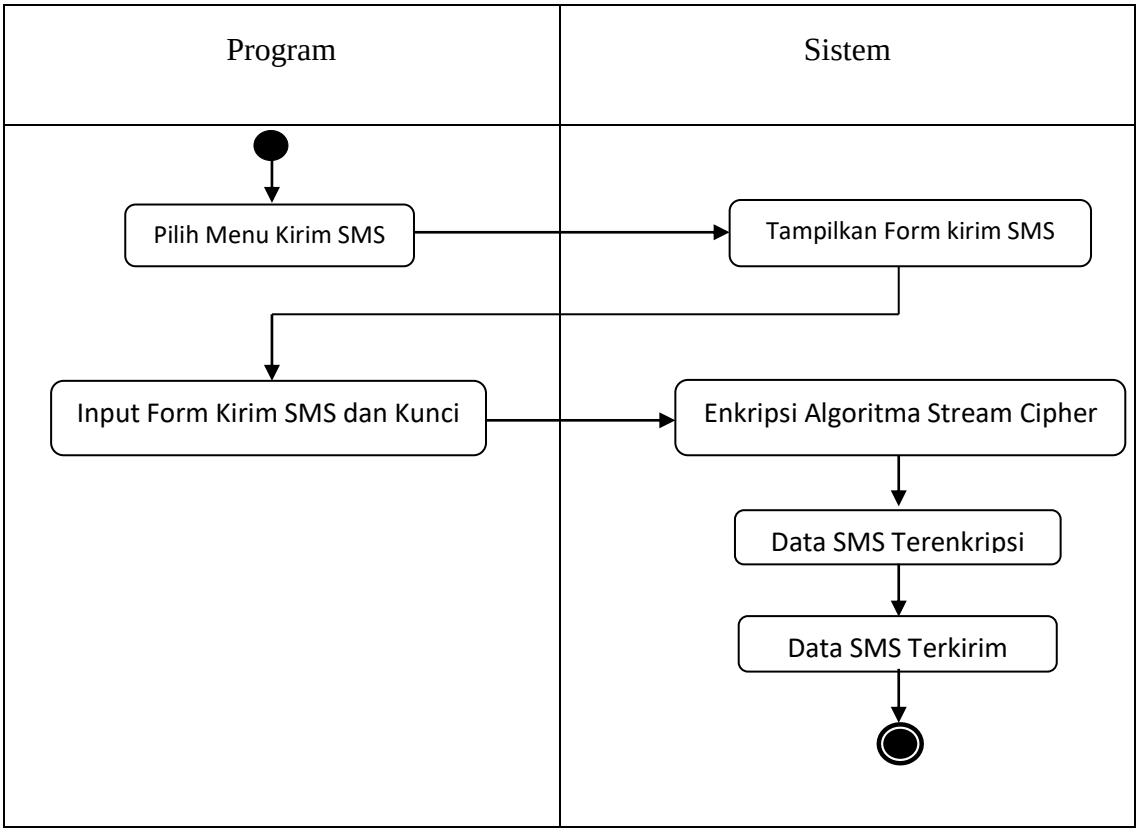
Sebuah *use case* mereprentasikan sebuah interaksi antara aktor dengan suatu sistem.



Gambar 3.2 Use Case

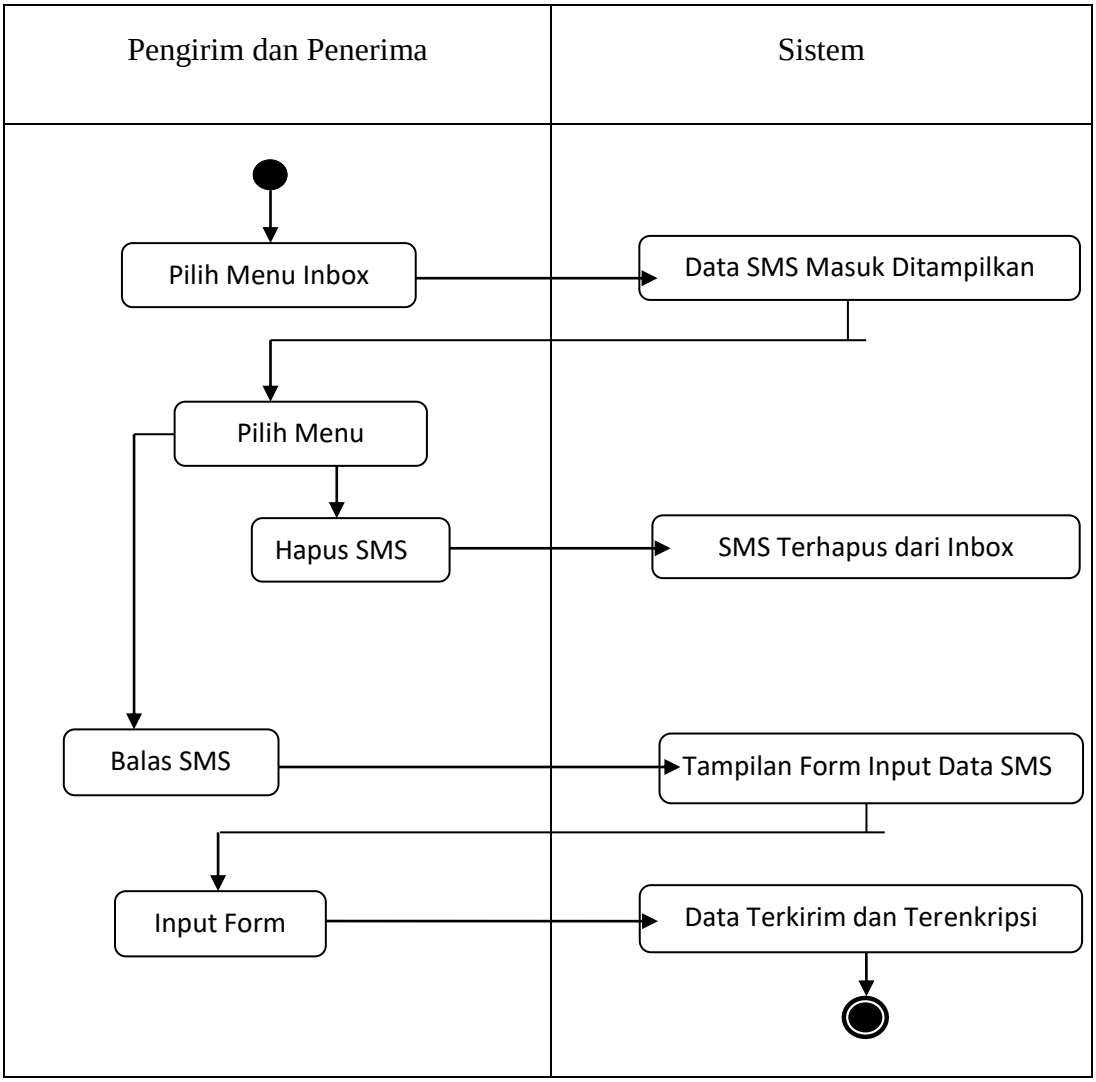
3.9.2 *Activity* Diagram

Activity diagram memodelkan alur kerja (*work flow*) sebuah urutan aktivitas pada suatu proses. *Activity* diagram dibuat untuk menggambarkan aktivitas aktor, dapat digambarkan sebagai berikut:



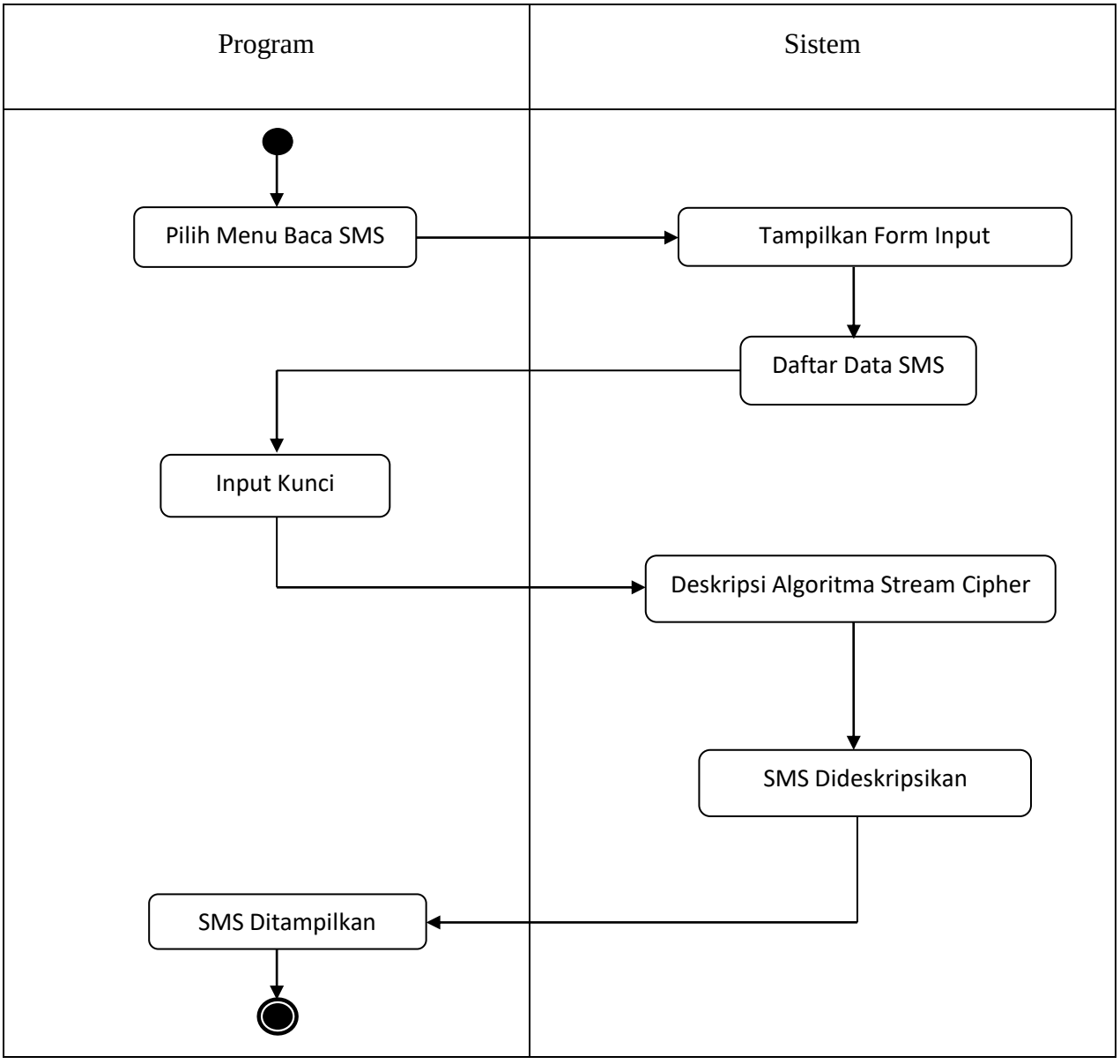
Gambar 3.3 *Activity* Diagram kirim SMS

3.9.3 *Activity* Diagram *Inbox* SMS



Gambar 3.4 *Activity* Diagram *Inbox* SMS

3.9.4 *Activity* Diagram Terima SMS

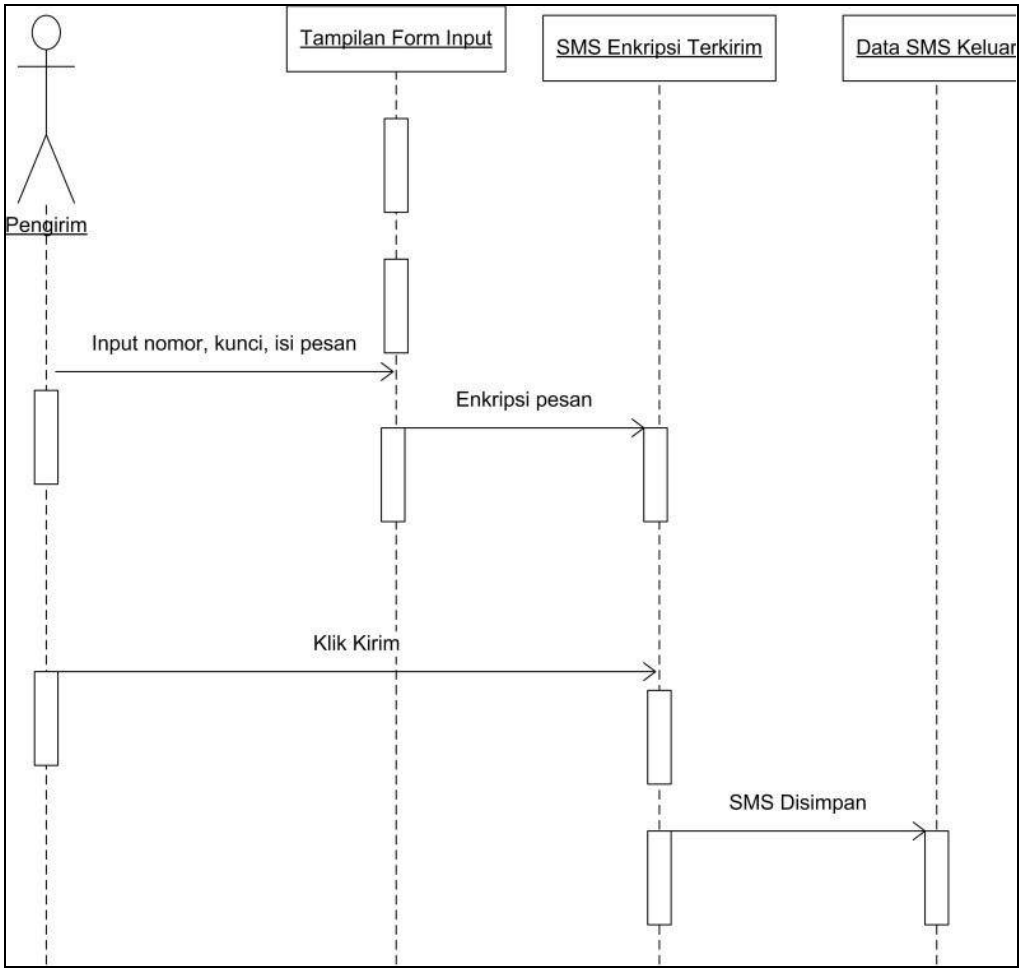


Gambar 3.5 *Activity* Diagram Terima SMS

3.9.5 *Sequence* Diagram

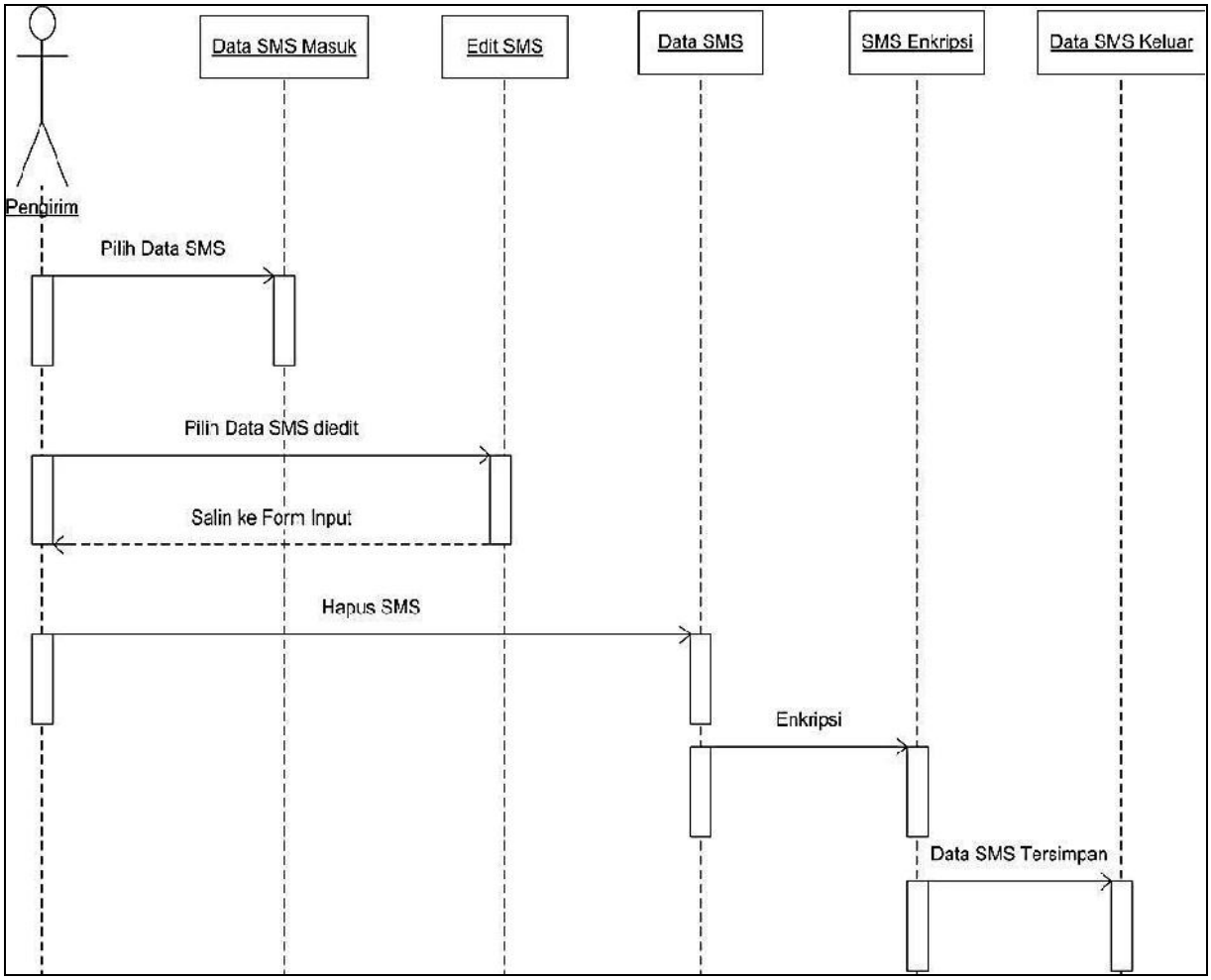
Sequence Diagaram menggambarkan interaksi antar objek didalam dan disekitar sistem berupa *message* yang digambarkan (termasuk penggunaan, *display*, dan sebagainya). Dapat kita lihat pada gambar berikut :

1. *Sequence* Diagram kirim SMS



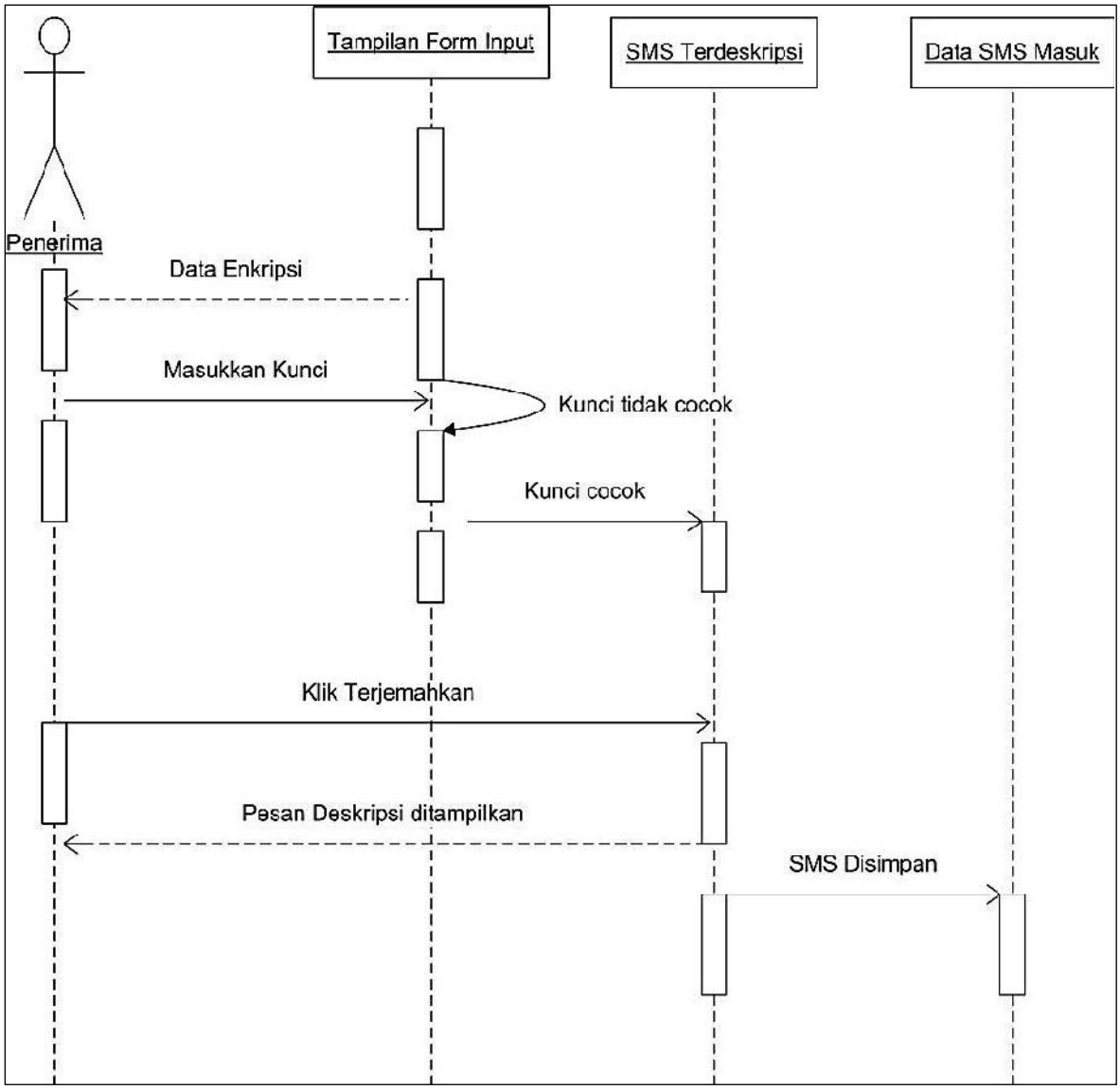
Gambar 3.6 *Sequence* Diagram kirim SMS

2. *Sequence* Diagram Kotak Masuk *(inbox)*



Gambar 3.7 *Sequense* Diagram Kotak Masuk *(inbox)*

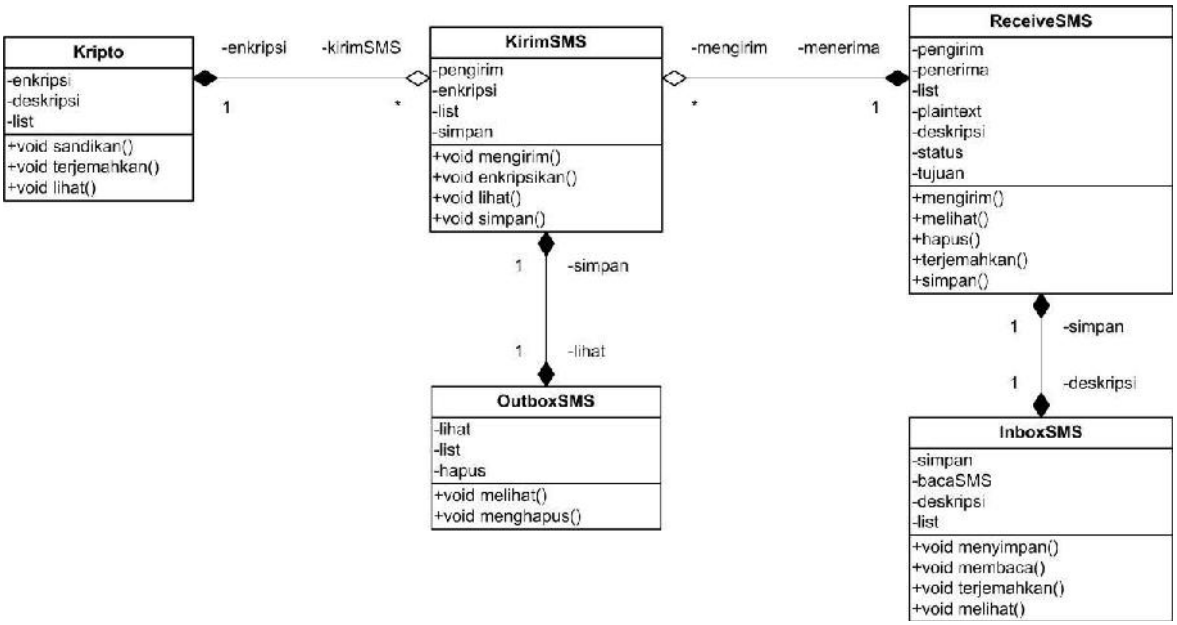
3. *Sequense* Diagram Terima SMS



Gambar 3.8 *Sequense* Diagram terima SMS

3.9.6 *Class Diagram*

Class Diagram adalah diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram dapat menunjukkan hubungan antar *class* dalam sistem yang sedang di bangun dan bagaimana saling berkolaborasi, dapat kita lihat pada gambar berikut :



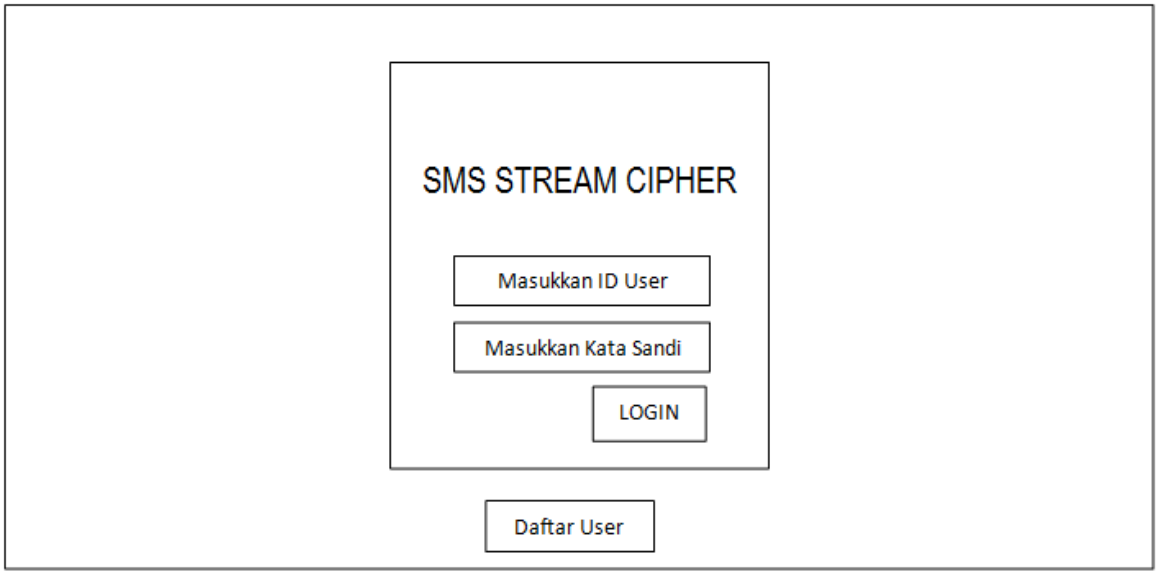
Gambar 3.9 *Class Diagram*

3.10 Rancangan Tampilan

Pada tahapan ini penulis menjelaskan rancangan tampilan halaman yang akan dibangun pada aplikasi yang direncanakan. Adapun rancangan tampilan masing-masing halaman *form* tesebut dapat dijelaskan sebagai berikut.

1. Perancangan awal

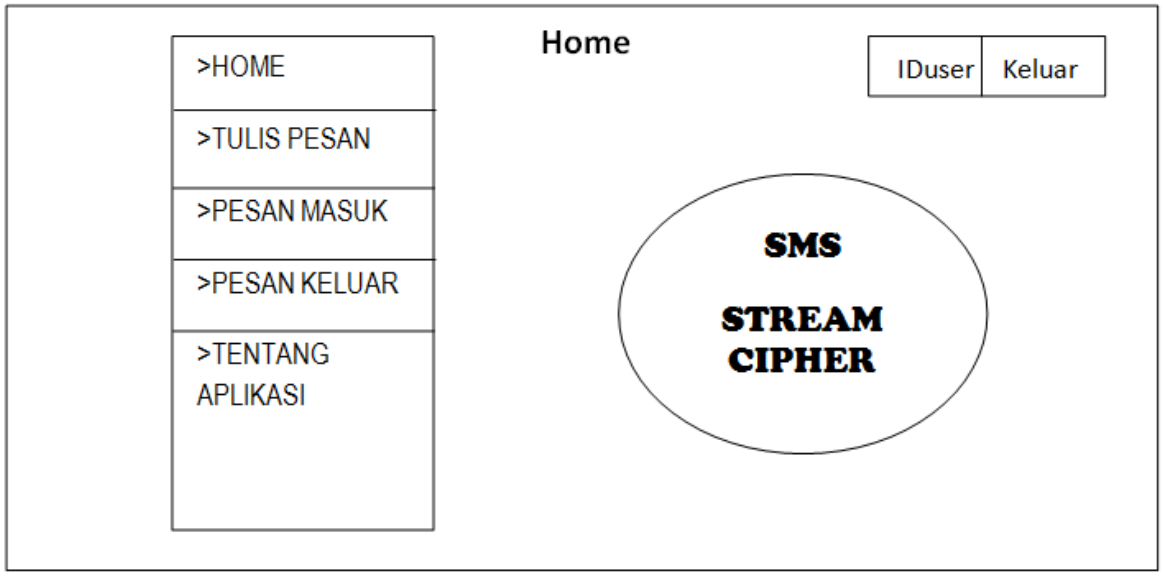
   Rancangan tampilan awal merupakan rancangan layar splash dimana saat aplikasi dibuka atau dijalankan, didalam tampilan ini dibuat untuk para pengguna melakukan daftar akun ataupun *Login* langsung bila sudah ada akun aplikasi. Adapun rancangan bisa dilihat berikut :



Gambar 3.10 Rancangan Tampilan Awal

2. Rancangan Tampilan Menu

   Rancangan Menu merupakan tampilan menu yang ada setelah pengguna masuk ke dalam aplikasi, dapat dilihat dibawah ini.



Gambar 3.11 Rancangan tampilan Menu

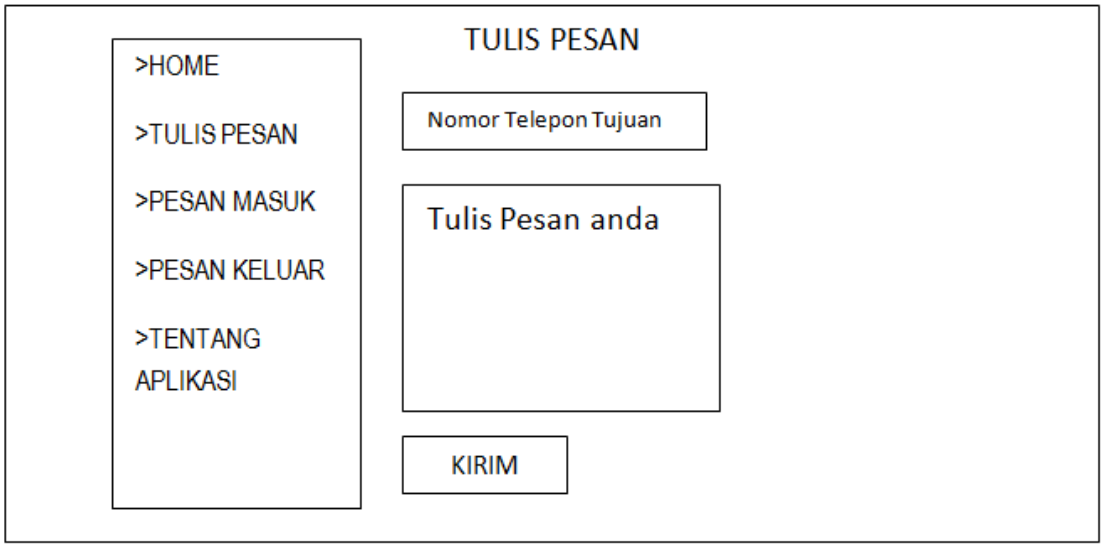
Pada gambar diatas terdapat menu yang ditampilkan dan dapat dijelaskan antara lain sebagai berikut :

1. Tulis Pesan, adalah menu masuk untuk merancang dalam membuat sms baru yaitu dengan mengetikkan text kata-kata melalui menu pesan.
2. Pesan Masuk, pada menu ini merupakan menu yang digunakan untuk membuka pesan masuk dan membaca pesan yang masuk.
3. Pesan Keluar, adalah menu yang diperuntukan membuka pesan yang terkirim
4. Tentang Aplikasi, ialah menu untuk menyajikan informasi mengenai *developer* pembuat program ini.

5. *IDuser*, adalah menu yang menampilknan nama dari akun yang sedang beroperasi *(online)*
6. Keluar, merupakan menu untuk kita keluar dari aplikasi

3. Rancangan Menu Tulis Pesan

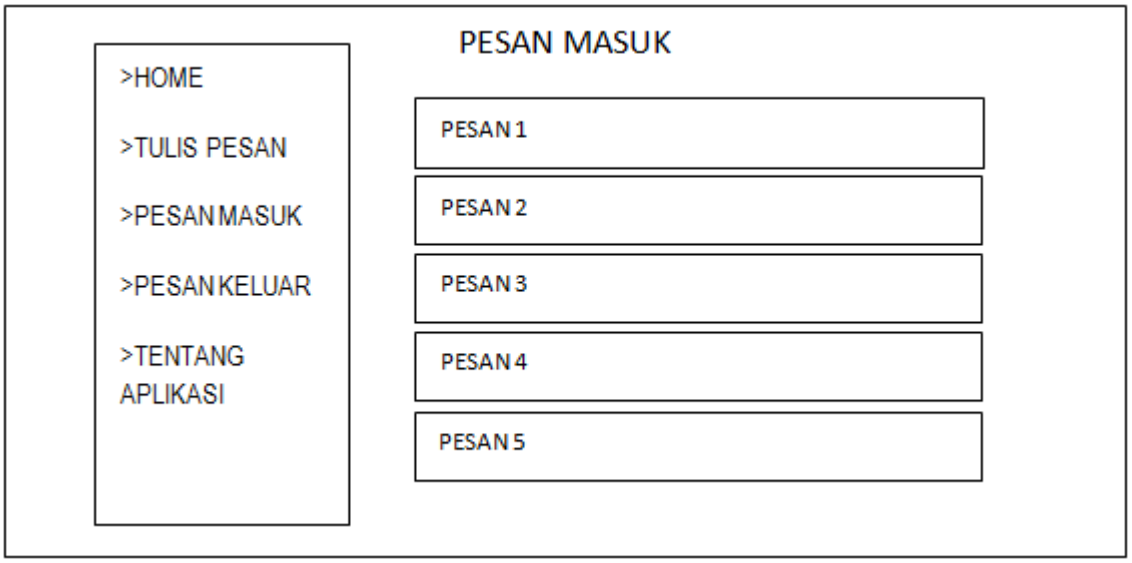
Rancangan Pesan Baru merupakan rancangan sebelum masuk ke form tulis pesan atau sms dijalankan, adapun rancangan bisa dilihat berikut :



Gambar 3.12 Rancangan Tampilan Tulis Pesan

4. Rancangan Pesan Masuk

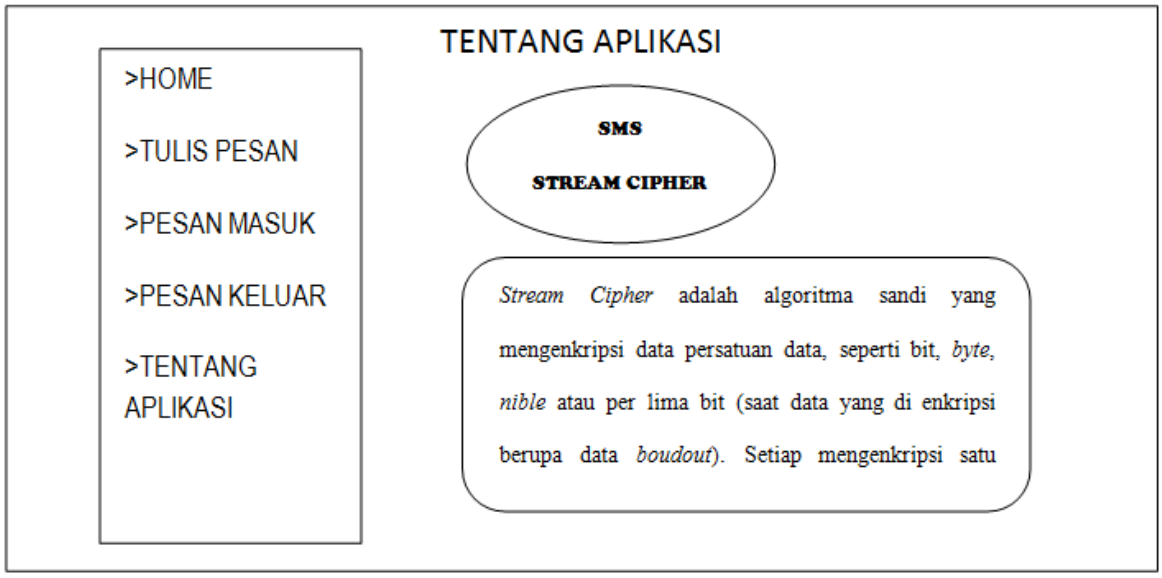
Rancangan Pesan Masuk berfungsi untuk menampilkan sms atau pesan yang masuk, dan pesan yang tersimpan, adapun rancangan bisa dilihat berikut :



Gambar 3.13 Rancangan Tampilan Pesan Masuk

5. Rancangan Tentang Aplikasi

Rancangan Tentang Aplikasi untuk menampilkan informasi tentang aplikasi dan info dari *developer* pembuat program, rancangan bisa dilihat berikut :



Gambar 3.14 Rancangan Tampilan Tentang Aplikasi

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini akan dijelaskan tampilan hasil dari aplikasi yang telah dibuat, yang digunakan untuk memperjelas tentang tampilan-tampilan yang ada pada aplikasi *SMS Stream Cipher* ini. Sehingga hasil implementasi dapat dilihat sesuai dengan hasil program yang telah dibuat.

Tahapan implementasi yang dilakukan untuk menyelesaikan perancangan sistem pengamanan *database SMS* yang menggunakan *Stream Cipher* ini diperlukan informasi mengenai penyediaan perangkat keras *(Hardware)* dan penyediaan perangkat lunak *(Software).*

4.1.1 Spesifikasi Perangkat Keras *(Hardware)*

Perancangan sistem pengamanan *database Short Message Service* yang menggunakan *Stream Cipher*, telah diuji pada laptop dengan spesifikasi perangkat keras sebagai berikut:

1. Prosessor : Intel(R) Core(TM) i3 CPU M380 @ 253GHz
2. Harddisk Space : 500 GB
3. Memory RAM : 2.00 GB
4. Monitor LCD 14 Inch
5. Keyboard

4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan untuk mendukung pembuatan program sistem pengamanan *database Short Message Service* yang menggunakan *Stream Cipher*. Adapun perangkat lunak yang digunakan sebagai berikut :

1. *Microsoft Windows 7*
2. *Mozila Firefox sebagai Browser*
3. *Database, XAMPP , PHP*, dan *MYSQL*
4. Aplikasi *SMS Gateway* pada *Smartphone* sebagai modem

4.1.3 Kebutuhan Pengguna

Pengguna adalah sumber daya manusia yang nantinya berperan sebagai *user administrator*. Pengguna yang dibutuhkan nantinya bekerja pada *server* memastikan berjalan baik untuk melayani permintan dari *customer* yang mengakses *server* tersebut. Penguna harus mengerti sistem dan mampu mengunakan komputer dan mengerti tentang prosedur yang sedang berjalan.

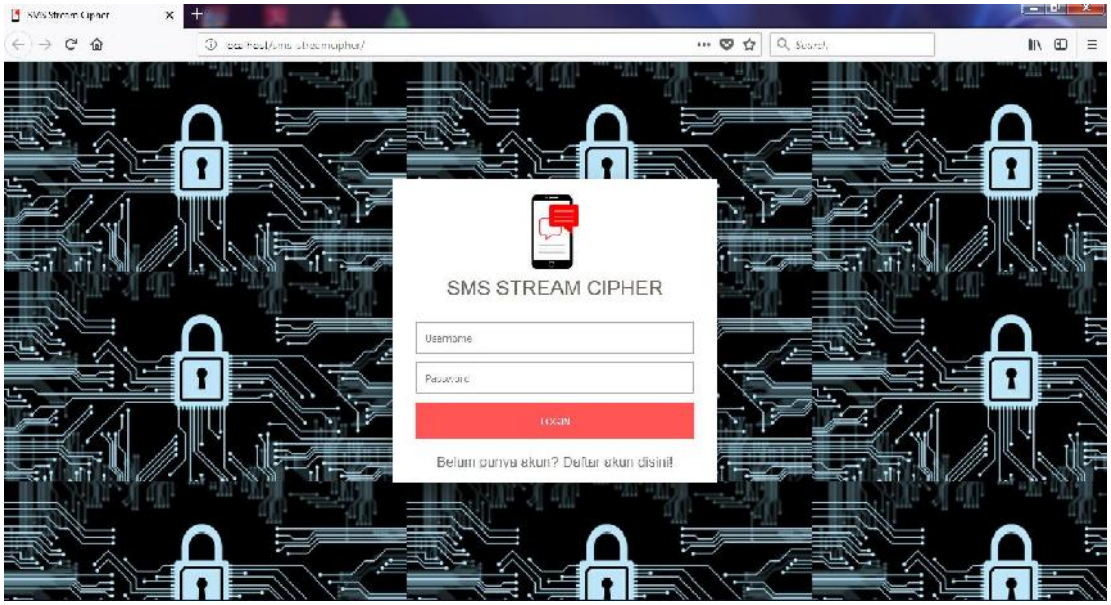
4.2 Implementasi Antar Muka

Implementasi antar muka dilakukan pada setiap halaman aplikasi yang sudah selesai dibuat dalam bentuk *file* program. Implementasi rancangan antarmuka dengan menggunakan *XAMPP* dan *PHP*. Implementasi sistem merupakan beberapa contoh form/ halaman tampilan ketika *user* memanfaatkan fasilitas yang tersedia pada sistem, tampilan sistem pengamanan data SMS menggunakan *Stream Cipher* terdiri atas tampilan *login,* tampilan halaman utama (home), tampilan tulis pesan, tampilan pesan masuk, tampilan pesan keluar, dan

tampilan tentang aplikasi. Adapun tampilan menu menu aplikasi pengamanan database sebagai berikut :

1. Tampilan *Login*

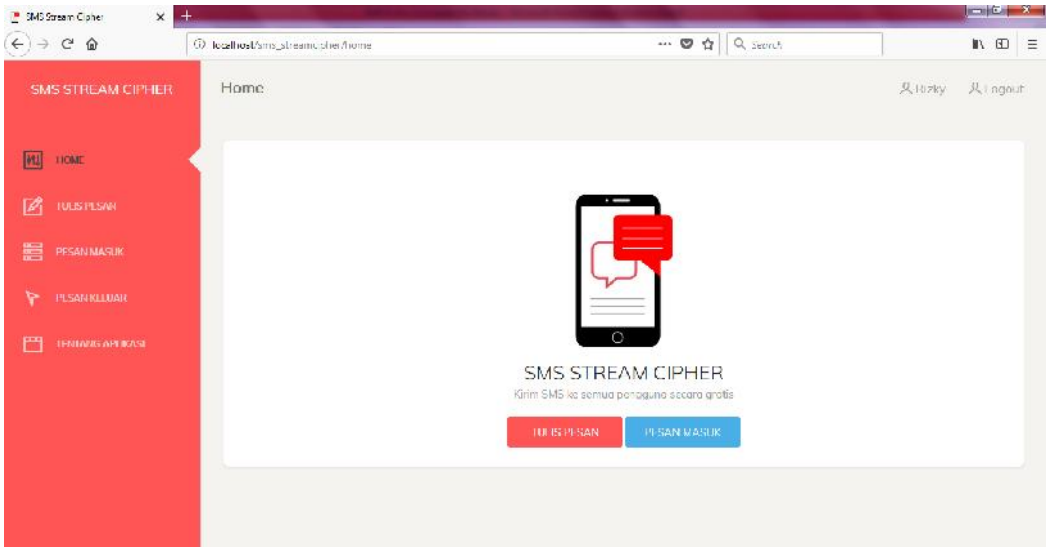
Tampilan *Login* digunakan untuk memasukan *username* dan *password*. Tampilan login merupakan tampilan yang pertama kali ditemukan ketika mengakses sistem.



Gambar 4.1 Tampilan Login

2. Tampilan *Home*

Tampilan *Home* merupakan tampilan menu-menu yang ada setelah pengguna masuk ke dalam aplikasi.



Gambar 4.2 Tampilan *Home*

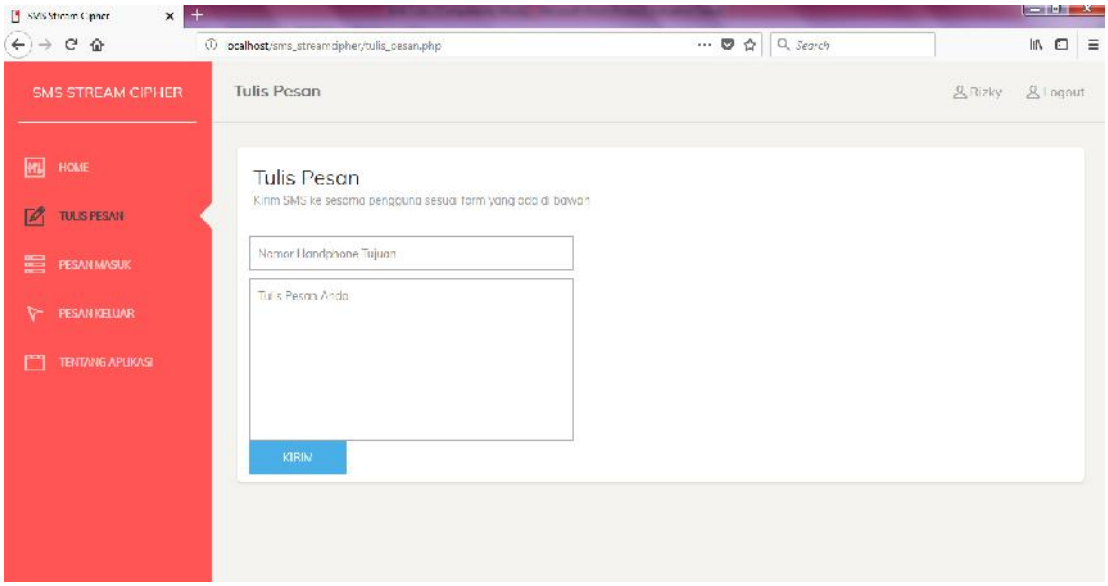
Pada gambar diatas terdapat menu yang ditampilkan dan dapat dijelaskan antara lain sebagai berikut :

1. Tulis Pesan, adalah menu masuk untuk merancang dalam membuat sms baru yaitu dengan mengetikkan kata-kata melalui menu pesan.
2. Pesan Masuk, pada menu ini merupakan menu yang digunakan untuk membuka pesan masuk dan membaca pesan yang masuk.
3. Pesan Keluar, adalah menu yang diperuntukan membuka pesan yang terkirim
4. Tentang Aplikasi, ialah menu untuk menyajikan informasi mengenai *developer* pembuat program ini.

5. *IDuser*, adalah menu yang menampilknan nama dari akun yang sedang beroperasi *(online)*
6. *Logout*, merupakan menu untuk kita keluar dari aplikasi

3. Tampilan Tulis Pesan

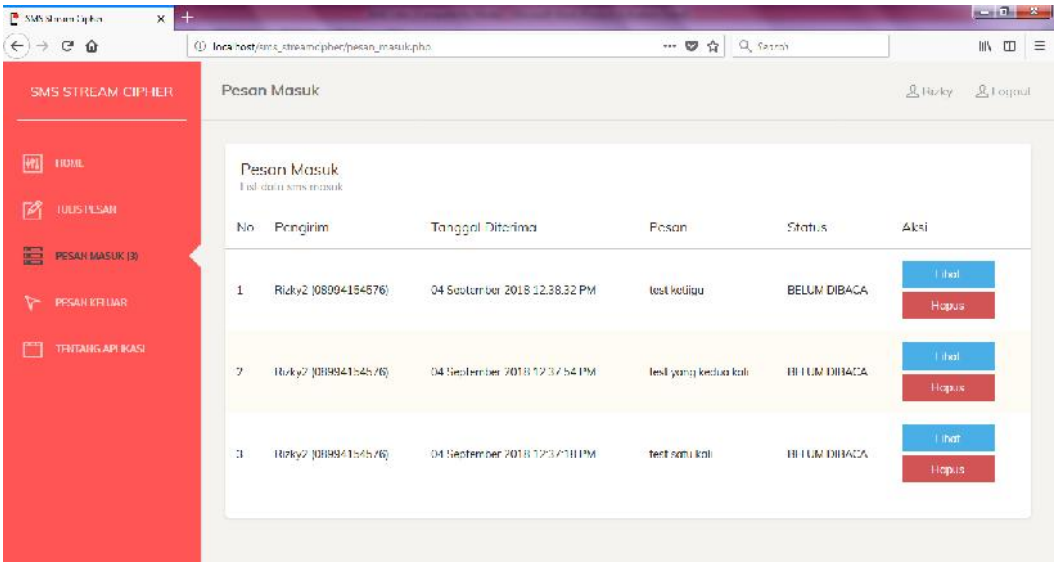
Tampilan tulis pesan ini merupakan tampilan dari menu masuk ke *form* tulis pesan atau sms dijalankan, adapun tampilan bisa dilihat berikut:



Gambar 4.3 Tampilan Tulis Pesan

4. Tampilan Pesan Masuk

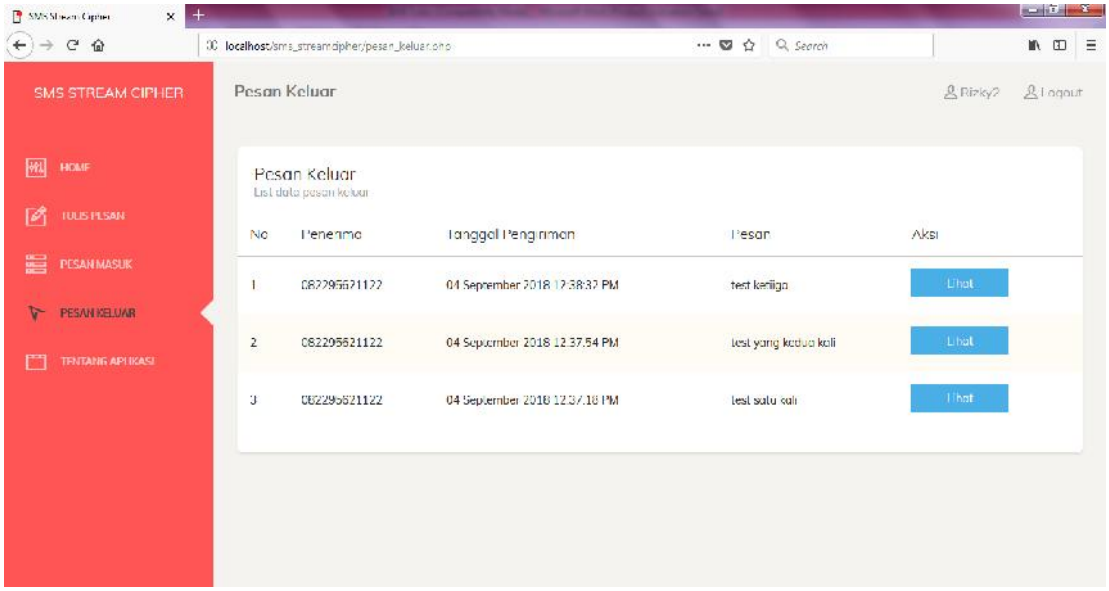
Pada menu ini menampilkan sms atau pesan yang masuk, dan pesan yang tersimpan.



Gambar 4.4 Tampilan Pesan Masuk

5. Tampilan Pesan Keluar

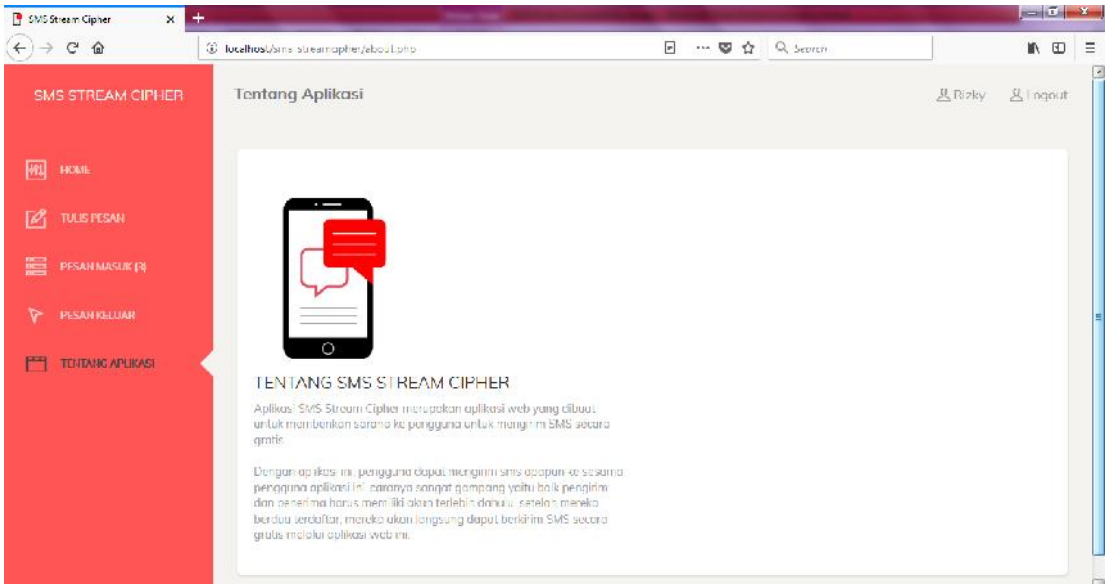
Tampilan menu yang diperuntukan membuka pesan yang terkirim sebelumnya, dibisa kita lihat sebagai berikut.



Gambar 4.5 Tampilan Pesan Keluar

6. Tampilan Tentang Aplikasi

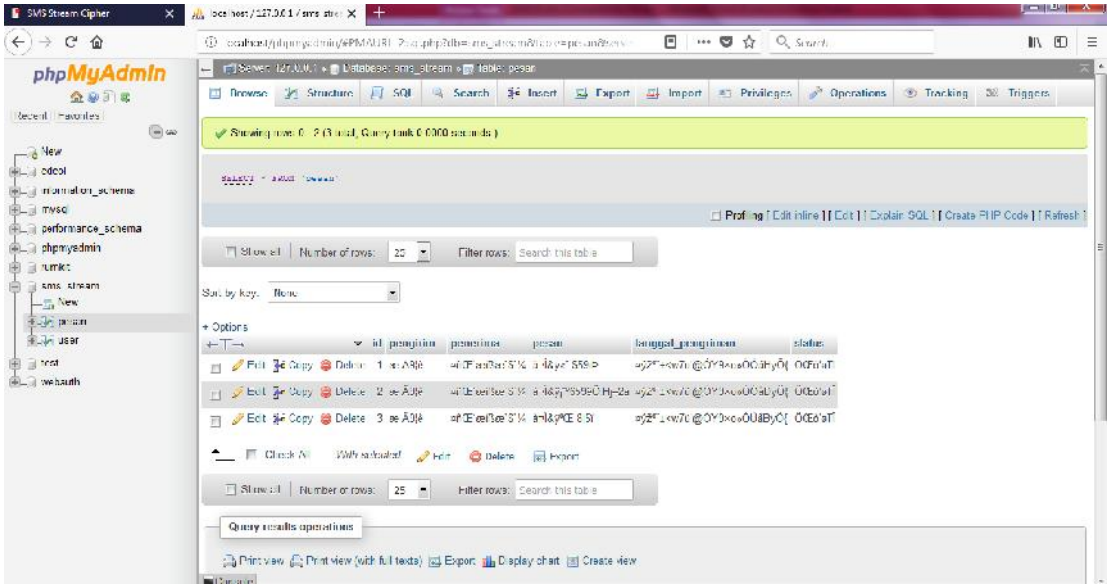
Tentang Aplikasi, ialah menu untuk menyajikan informasi mengenai *developer* pembuat program ini.



Gambar 4.6 Tampilan Tentang Aplikasi

7. Tampilan *Database* Yang Sudah di Enkripsi Dengan *Stream Cipher*

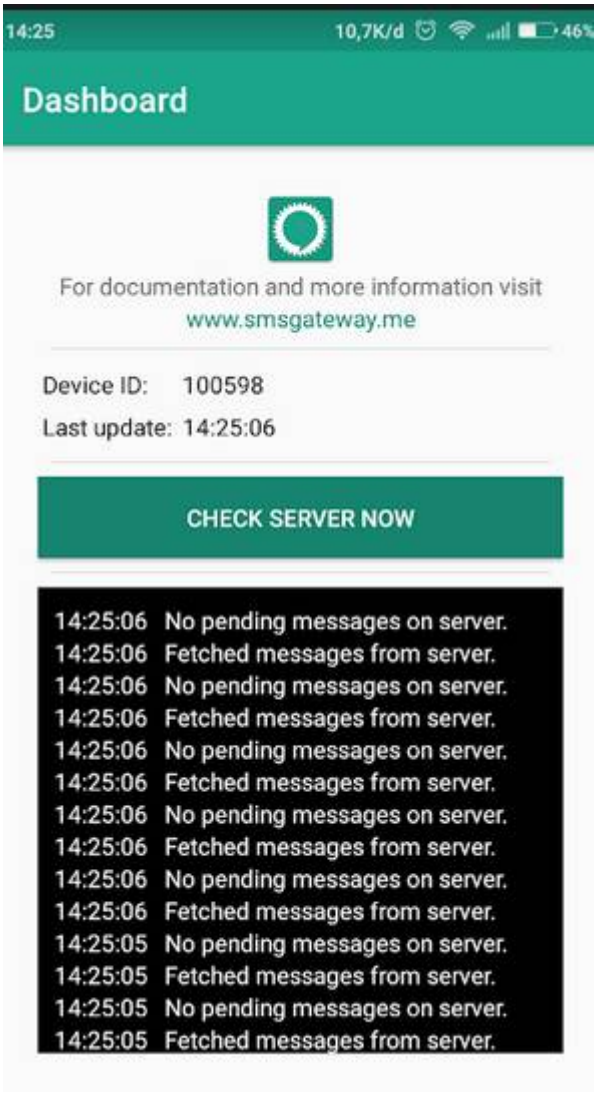
Tampilan data yang di dalam *database* ini berisi tentang data di aplikasi, baik itu data *user* maupun data pesan yang tersimpan dan yang terkirim akan tetapi data ini sudah diubah dan diamankan menggunakan *Stream Cipher* seperti pada gambar dibawah ini.



Gambar 4.7 Tampilan *Database* Yang Sudah Enkrip dengan *Stream Cipher*

8. Tampilan Dari Aplikasi SMS *Gateway* di *Smartphone*

SMS *Gateway* yang berfungsi sebagai Modem dimana penghubung antara *website* dengan penerima SMS. Dan dibawah ini adalah tampilan yang memberitahu kita SMS telah sampai ke nomor tujuan.



Gambar 4.8 Tampilan Aplikasi SMS *Gateway*

4.3 Pengujian

Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibuat memiliki kualitas yang baik dan terbukti berjalan dengan baik, yaitu mampu mempersentasikan kajian pokok dari spesifikasi analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Pengujian aplikasi ini menggunakan metode pengujian *Black Box*. Pengujian dengan *black box* berfokus pada persyaratan fungsional perangkat lunak.

Tabel 4.1 Hasil Pengujian *Black Box Testing*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Membuka Halaman Awal Aplikasi	Loading Web	Aplikasi memproses Loading Form Login dan Menuju Ke Menu Utama	Sesuai dengan yang diharapkan	Valid
2	Proses Enkripsi Sms	*Send Sms*	Ketika SMS yang telah ditullis dan dikirim ke penerima pesan sudah di enkripsi	Sesuai dengan yang diharapkan	Vallid
3	Menerima Pesan, Melihat Pesan, Terkirim, Simpan Pesan	Form Inbox/ Outbox	Aplikasi bisa menerima pesan yang dikirimkan melalui perangkat lain, ketika pengiriman pesan berhasil aplikasi dapat menampilkan history pesan, aplikasi dapat menyimpan pesan yang dinginkan pengguna	Sesuai dengan yang diharapkan	Valid

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan dari sistem yang telah dibuat dan dirancang adalah rangkuman dari semua hasil penelitian yang telah dilakukan oleh penulis, beberapa kesimpulan yang akan berguna untuk kedepannya agar dapat dikembangkan ataupun diberi masukkan bagi para pembaca dan penulis ataupun untuk masyarakat umum. Adapun beberapa kesimpulan berdasarkan keseluruhan proses yang dilakukan untuk membuat Aplikasi Sistem Kriptografi dalam Pengamanan *Database* SMS dengan menggunakan *Stream Cipher* sebagai berikut:

1. Aplikasi sistem kriptografi yang dirancang telah berhasil dilaksanakan dan dapat berjalan dengan baik dalam pengamanan pada *database* SMS dengan menggunakan *Stream Cipher RC4*. Perancangan aplikasi dilakukan dengan cara menggunakan diagram *UML* yaitu *Use Case* Diagram, *Activity* Diagram, *Squence* diagram, dan *Class* diagram.
2. Aplikasi yang dirancang ini berbasis sistem kriptografi yang dapat menghindari penyadapan pada *database* SMS atau pesan teks dari pihak yang tidak bertanggung jawab.
3. Tingkat keamanan dari sistem ini bisa dikatakan cukup baik ataupun standart, karena telah diuji dengan menggunakan pengujian *Black Box*.

5.2 Saran

Ada beberapa saran berdasarkan keseluruhan proses yang dilakukan untuk membuat Aplikasi Sistem Kriptografi dalam Pengamanan *Database* SMS dengan menggunakan *Stream Cipher*, untuk pengembangan lebih lanjut maka penulis memberikan saran yang sangat bermanfaat dan membantu dalam pengamanan *database* pesan teks untuk masa yang akan datang, yaitu:

1. Dalam aplikasi ini masih dilakukan pengamanan pada pesan teks SMS, sehingga sangat dibutuhkan pengembangan aplikasi sejenis yang berguna dalam mengenkripsi ataupun melakukan pengamanan *database* pada file sejenis pesan teks.
2. Pada tampilan desain dapat dimanfaatkan oleh pengembang untuk membuat desain yang lebih menarik perhatian masyarakat umum sehingga menambah minat para pengguna dalam mengembangkan aplikasi ini.
3. Untuk penggunaan aplikasi ini disarankan menggunakan koneksi yang tinggi agar pesan yang dikirim dan diterima dapat sampai tujuan dengan cepat.
4. Aplikasi ini agar kiranya dapat digunakan secara baik supaya menghasilkan suatu tujuan yang baik pula bagi yang menggunakanya.

DAFTAR PUSTAKA

Anonim, E. H. Rachmawanto and C. A. Sari, "Keamanan File Menggunakan Teknik Kriptografi Shift Cipher," Jurnal Techno. Com, vol. 14, no. 2, pp. 329-335, 2014.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11

Bishop, Rosdiana, "Sekuritas Sistem Dengan Kriptografi," in Prosiding Sendi_U 2013, Semarang, 2013.

FACHRI, Barany. Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 2018, 3: 98-102.

Fresly, Faizal Zuli1, Ari Irawan, "Implementasi Kriptografi Dengan Algoritma Blowfish dan Riverst Shamir Adleman (RSA) Untuk Proteksi File," Jurnal Format Volume 6 nomor 2 Tahun 2016.

Gede Angga Pradipta " Penerepan Kombinasi metode Enkripsi Vigenere Cipher Dan Trasposisi Pada Aplikasi Client Server Chatting, " Jurnal Sistem Dan Informatika Vol. 10, Nomor 2, 2016.

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). Implementasi Augmented Reality Sebagai Media Promosi Penjualan Rumah. In Seminar Nasional Royal (Senar) (Vol. 1, No. 1, pp. 429-434).

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Nandar Pabokory, Indah Fitri Astuti, Awang Harsa Kridalaksana, " Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," Jurnal Informatika Mulawarman Vol. 10. Nomor 1, 2015.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." INTECOMS: Journal of Information Technology and Computer Science 1.1 (2018): 72-77.

Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.

Ramadhan, A., & Mohd. Awal Hakimi. (2006). Pemprograman Web Database dengan PHP dan MySQL. Synergy Media.

Ramadhan, M., & Nugroho, N. B. (2009). Desain web dengan php. Jurnal Saintikom, 6(1).

Renddy, Teady Matius, Surya Mulyana, Fresly, " Steganografi Dengan Deret Untuk Mengacak Pola Penempatan Pada Rgb," Jurnal Teknologi Informasi, 2015.

Rhee, C. A. Sari, E. H. Rachmawanto, Y. P. Astuti and L. Umaroh, "Optimasi Penyandian File Kriptografi Shift Cipher," in Prosiding Sendi_U 2013, Semarang, 2013.

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.

Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.

Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." Int. J. Eng. Trends Technol 42.5 (2016): 218-222.

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Suriski Sitinjak, Yuli Fauziah, Juwairiah, " Aplikasi Kriptografi File Menggunakan Algoritma Blowfish," Jurnal Informatika Mulawarman Vol. 10. Nomor 1, 2015.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.