IMPLEMENTASI ALGORITMA *RIVEST CODE* 6 (RC6) UNTUK ENKRIPSI DAN DEKRIPSI PESAN SMS PADA PONSEL BLACKBERRY

Nopiyanto
Program Studi Manajemen Informatika, STMIK Cikarang
E-mail : info.nopiyanto@gmail.com

ABSTRAKSI

Pertukaran pesan dengan cepat pada era globalisasi ini sangat dibutuhkan untuk memenuhi kebutuhan manusia. BlackBerry menyediakan fasilitas *Short Message Service* (SMS) yang merupakan media yang memungkinkan penggunanya untuk berkomunikasi dalam bertukar pesan dari pengirim ke penerima informasi dengan biaya yang relatif murah. Zaman yang semakin berkembang, maka perlunya suatu sistem yang dapat mengamankan informasi tersebut menjadi lebih aman agar tidak terjadinya penyadapan yang menyebabkan pesan tersebut tidak aman. Kriptografi merupakan suatu cabang ilmu yang dapat mengamankan pesan dengan algoritma tertentu, dengan menggunakan algoritma kriptografi maka pesan tersebut akan lebih aman sehingga peluang untuk menyadap pesan tersebut menjadi lebih kecil. Algoritma yang baik dapat memperkuat sistem dalam mengamankan pesan *Short Message Service* (SMS), algoritma *Rivest Code* 6 (RC6) merupakan salah satu algoritma kriptografi yang dapat mengamankan pesan *Short Message Service* (SMS) pada ponsel BlackBerry. Dengan teknik enkripsi dan dekripsi maka pertukaran pesan menjadi lebih mudah dalam mengamankan pesan tersebut. Tujuan dari pengembangan aplikasi ini yakni membantu masyarakat dalam mengamankan pesan *Short Message Service* (SMS), sehingga masyarakat dapat meminimalisir proses kejahatan seperti penyadapan pesan Short Message Service (SMS) pada ponsel BlackBerry yang dibangun dengan menggunakan bahasa pemograman JAVA.

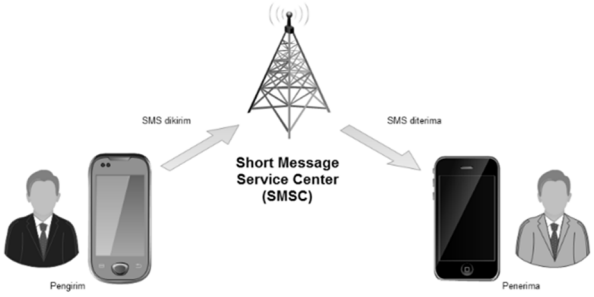
Kata Kunci : Algoritma Rivest Code 6 (RC6), BlackBerry, *Short Message Service* (SMS).

1. PENDAHULUAN

Menurut The Concise Oxford Dictionary (2006) mendefiniskan kriptografi sebagai seni menulis atau memecahkan kode (1). Dari definisi tersebut dapat disimpulkan bahwa kriptografi merupakan suatu cabang ilmu yang dapat membuat kode atau memecahkan kode sehingga dapat mengamankan sesuatu, baik data maupun informasi dengan proses enkripsi dan dekripsi.

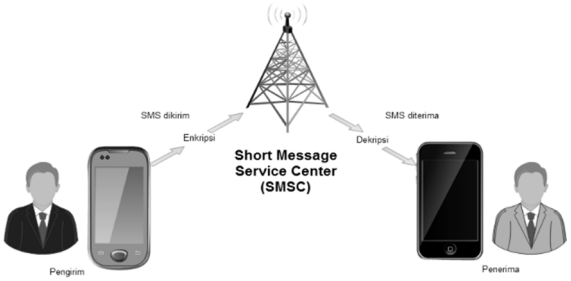
BlackBerry pada saat ini merupakan ponsel yang paling banyak digunakan oleh sebagian orang di dunia. BlackBerry menyediakan sarana komunikasi yang beragam dan salah satunya SMS, SMS merupakan layanan yang memunginkan penggunanya untuk melakukan pengiriman pesan singkat kepada orang yang ditujunya dengan cepat dan biaya yang murah.

Layanan SMS yang menggunakan aplikasi SMS bawaan ponsel masih banyak digunakan oleh masarakat, dan bukan jalur yang aman dalam pertukaran informasi. Pesan yang yang dikirim dengan aplikasi SMS bawaan ponsel masih berupa teks yang terbuka yang belum terproteksi selain itu pengiriman pesan tidak langsung diterima oleh penerima secara langsung melainkan pengiriman SMS harus melewati *Short Message Service Center* (SMSC) yang berfungsi mencatat komunikasi pesan antara pengirim dan penerima. Dengan tersimpannya pesan di SMSC maka seorang operator dapat melihat pesan dikirim oleh pengirim ke penerima, dengan terjadinya hal tersebut maka penyadapan tidak bisa hindari.



Gambar 1.1 Arsitektur global pengiriman pesan SMS tanpa keamanan

Oleh karena itu perlunya algoritma untuk diimplementasikan dalam perangkat lunak yang dibangun di atas platform Blakberry yang dapat membantu pengiriman pesan melalui SMS pada ponsel BlackBerry menjadi lebih aman. Algoritma *Rivest Code* 6 (RC6) menjadi pilihan karena algoritma tersebut merupakan salah satu kandidat *Advanced Encryption Standard* (AES) (2) yang merupakan peyempurna dari algoritma *Data Encryption Standard* (DES) yang sudah tidak aman lagi untuk diimplementasikan.



Gambar 1.2 Arsitektur global pengiriman pesan SMS dengan keamanan

2. LANDASAN TEORI

2.1 Algoritma *Rivest Code* 6 (RC6)

Algoritma *Rivest Code* 6 (RC6) merupakan algoritma kriptografi yang menggunakan kunci yang sama dalam proses enkripsi dan dekripsi pesan (simetri), didesain oleh Ronal Linn Rivest, Matt J.B Robshaw, Ray Sydney, dan Yuqin Lisa Yin dari laboratorium RSA. Dilihat dari singkatan kata RC6 mempunyai arti sebagai "*Rivest Code*" sedangkan 6 merupakan versi dari algoritma sebelumnya seperti *Rivest Code* 4 (RC4), *Rivest Code* 5 (RC5).
Perbedaan antara algoritma RC4 dengan algoritma RC5, dan RC6 terletak pada banyaknya register yang tersedia. Tetapi algoritma RC4 mengolah data bukan bedasarkan blok (*block cipher*) seperti algoritma RC5, dan algoritma RC6, melainkan mengolah data bedasarkan aliran data (*stream cipher*). Penggunaan register dibutuhkan untuk penempatan blok data yang akan diproses, sehingga algoritma memproses data pada tiap register yang ada.

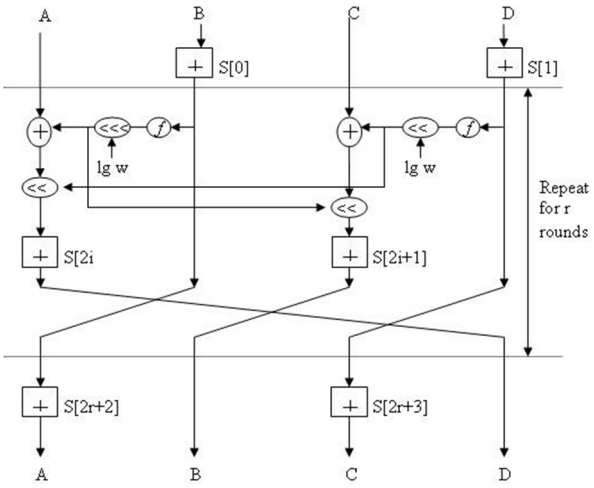
Algoritma RC6 merupakan algoritma yang sederhana, fungsi yang digunakan merupakan fungsi yang sederhana dan hanya mengandalkan prinsip *iterated cipher* untuk keamanan. Karena panjang karakter dari hasil enkripsi mempunyai panjang 8 bit, sedangkan sebagian telepon selular hanya dapat menampilkan karakter dengan panjang 7 bit. Oleh karena itu untuk mengatasi permasalahan tersebut, karakter-karakter yang akan dienkripsi diubah terlebih dahulu kedalam nilai ASCII (*American Code for Information Interchange*), dimana nilai karakter dalam tabel ASCII ditambah karakter spesial 0 sampai dengan 255. Artinya setiap satu karakter ASCII akan diwakili 8 bit, dimana 28 = 256, sehingga dalam 1 blok *plaintext* (32 bit) akan menyimpan 4 karakter dan setiap kali iterasi/perulangan akan diambil 16 karakter dari *plaintext.* Apabila panjang *plaintext* atau panjang kunci kurang dari 16 karakter, maka akan dilakukan *padding,* yaitu dengan menambahkan bit "0" (nol) di akhir teks, sehingga panjang teks mencukupi 16 karakter.

Algoritma RC6 mempunyai beberapa parameter, sehingga dituliskan sebagai RC6 *–w/r/b*. Dari ketiga parameter tersebut dapat dirincikan bahwa ukuran register *w* (32, 64, atau 128 bit), parameter *r* merupakan bilangan bulat bukan negatif yang menunjukan banyaknya iterasi/perulangan selama proses enkripsi dan dekripsi, sedangkan parameter *b* menunjukan ukuran kunci dalam ukuran byte.
Untuk mempermudah proses pengembangan perangkat lunak, maka dari parameter-parameter diatas akan ditetapkan bahwa parameter *w* mempunyai ukuran 32 bit, parameter *r* sebesar 20 kali iterasi/perulangan, sedangkan parameter *b* memiliki panjang kunci yang beragam lebih dari 1 karakter (8 bit).
RC6 -w/r/b memecah blok 128 bit menajdi 4 buah blok 32-bit, dan mengikuti aturan enam operasi dasar.

1. a + b operasi penjumlahan bilangan integer
2. a – b operasi pengurangan bilangan integer
3. a ⊕ b operasi *exclusive*-OR (XOR)
4. a x b operasi perkalian bilangan integer
5. a<<<b a dirotasikan ke kiri sebanyak variabel kedua (b)
6. a>>>b a dirotasikan ke kanan sebanyak variabel kedua (b)

2.2 Algoritma Enkripsi RC 6

Karena RC6 memecah blok 128 bit menjadi 4 buah blok 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. Byte yang pertama dari *plaintext* atau *ciphertext* ditempatkan pada byte A, sedangkan byte yang terakhirnya ditempatkan pada byte D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.



Gambar 2.1 Diagram Blok Proses Enkripsi RC 6

Algoritma RC6 menggunakan 44 buah sub kunci yang dibangkitkan dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing-masing sub kunci panjangnya 32 bit. Proses enkripsi pada algoritma RC6 dimulai dan diakhiri dengan proses *whitening* yang bertujuan untuk menyamarkan iterasi yang pertama dan yang terakhir dari proses enkripsi dan dekripsi. Pada proses *whitening* awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[i]. Pada masing-masing iterasi pada RC6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah

iterasi ke-20 selesai, dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43]. Setiap iterasi pada algoritma RC6 nilai B dimasukan ke dalam fungsi f, yang didefinisikan sebagai f(x) =x(2x+1), kemudian diputar kekiri sejauh lg-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C.

Nilai t juga digunakan sebagai acuan bagi C untuk memutar nilainya kekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan proses pemutaran kekiri. Kemudian sub kunci S[2i] pada iterasi dijumlahkan dengan A, dan sub kunci S[2i+1] dijumlahkan dengan C. Keempat bagian dari blok kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempakan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai (asli) D ditempatkan pada C. demikian iterasi tersebut akan terus berlangsung hingga 20 kali.

Encryption with RC6-$w/r/b$

Input: Plaintext stored in four w-bit input registers A, B, C, D
Number r of rounds
w-bit round keys $S[0, \ldots, 2r+3]$

Output: Ciphertext stored in A, B, C, D

Procedure:
$B = B + S[0]$
$D = D + S[1]$
for $i = 1$ **to** r **do**
{
$t = (B \times (2B+1)) \lll \lg w$
$u = (D \times (2D+1)) \lll \lg w$
$A = ((A \oplus t) \lll u) + S[2i]$
$C = ((C \oplus u) \lll t) + S[2i+1]$
$(A, B, C, D) = (B, C, D, A)$
}
$A = A + S[2r+2]$
$C = C + S[2r+3]$

Gambar 2.2 *Pseudocode* Enkripsi RC6

2.3 Algoritma Dekripsi RC 6

Proses dekripsi *ciphertext* pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses *whitening*, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses *whitening* setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses *whitening* sebelum iterasi pertama digunakan pada *whitening* setelah iterasi terkahir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.

Decryption with RC6-$w/r/b$

Input: Ciphertext stored in four w-bit input registers A, B, C, D
Number r of rounds
w-bit round keys $S[0, \ldots, 2r+3]$

Output: Plaintext stored in A, B, C, D

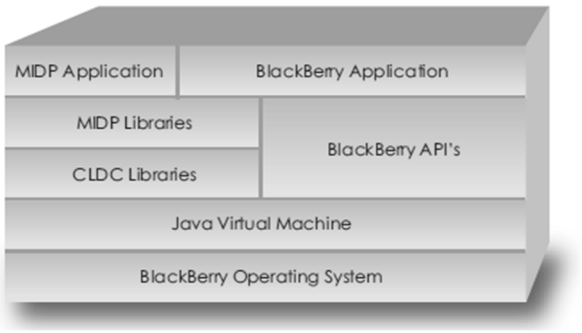
Procedure:
$C = C - S[2r+3]$
$A = A - S[2r+2]$
for $i = r$ **downto** 1 **do**
{
$(A, B, C, D) = (D, A, B, C)$
$u = (D \times (2D+1)) \lll \lg w$
$t = (B \times (2B+1)) \lll \lg w$
$C = ((C - S[2i+1]) \ggg t) \oplus u$
$A = ((A - S[2i]) \ggg u) \oplus t$
}
$D = D - S[1]$
$B = B - S[0]$

Gambar 2.2 *Pseudocode* Dekripsi RC6

2.4 Sistem Operasi BlackBerry

Sistem operasi BlackBerry merupakan milik sistem operasi mobile, yang dikembangkan oleh Research In Motion (RIM) untuk perusahaan BlackBerry garis smartphone perangkat genggam. Sistem operasi BlackBerry menyediakan *multitasking* dan mendukung perangkat input khusus yang telah diadopsi oleh RIM untuk digunakan dalam *handheld*, khususnya *trackwheel, trackball*, dan yang paling baru *trackpad* dan *touchscreen*.

Platform BlackBerry mungkin paling dikenal karena dukungan asli untuk *e-mail* perusahaan, melalui MIDP 1.0 dan, baru-baru ini, sebuah subset dari MIDP 2.0, yang memungkinkan aktivasi nirkabel lengkap dan sinkronisasi dengan Microsoft Exchange, Lotus Domino, atau Novell GroupWiseemail, kalender, tugas, catatan, dan kontak, bila digunakan dengan BlackBerry Enterprise Server.



Gambar 2.3 Arsitektur sistem operasi BlackBerry (9)

2.5 *Short Message Service* (SMS)

Short Message Service disingkat dengan SMS, merupakan pesan singkat berupa teks yang dikirim dan diterima antar sesama pengguna telepon, pada awalnya pesan ini digunakan antar telepon genggam, namun dengan

berkembangannya teknologi, pesan tersebut bisa dilakukan melalui komputer atau telepon rumah.

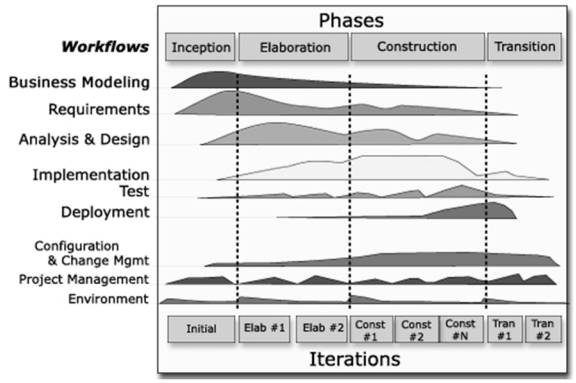
Arsitektur SMS terdiri dari *Short Message Entity* (SME), SMS *Service Center* (SMSC), dan *email Gateway* yang terhubung dengan elemen-elemen GSM maupun CDMA sebagai *channel* penghantar. SMS memiliki beberapa fitur dasar, yaitu:

1. *Message Submission and Delivery*
   a. *Message sending*: pesan dikirim dari MS (*Mobile Station*) ke SMSC (SMS *Service Center*), dialamatkan ke SME (*Short Message Entity*) lain. SME asal akan memeriksa apakah pesan yang dikirim masih berlaku. Jika tidak, maka SMSC akan menghapus pesan tersebut.
   b. *Message delivery*: SMSC akan menyampaikan pesan ke MS, dikenal dengan nama *Short message Mobile Terminated* (SM-MT).
2. *Status Report*: Status ini akan diminta oleh SME asal untuk mengetahui apakah pesan yang dikirim sukses atau tidak.
3. *Reply Path*: diatur oleh SME asal atau SMSC *serving* agar bisa menangani balasan SME penerima.

2.6 Proses Pengembangan Perangkat Lunak

Dalam membangun suatu perangkat lunak diperlukan sebuah metode, metode tersebut bertujuan untuk memeriksa suatu masalah, memahami dan mencari solusi. Bukan hanya metode saja yang diperlukan dalam membangun suatu perangkat lunak, karena metode tidak dapat terealisasi tanpa adanya suatu teknik. Teknik diperlukan sebagai alat bantu yang digunakan untuk memodelkan hasil analisis kedalam model yang dimengerti oleh pengembang perangkat lunak lainnya.

Proses pembangunan suatu aplikasi ini harus menggabungkan strategi pengembangan yang meliputi lapisan proses, metode dan alat-alat bantu. Strategi ini disebut juga dengan model proses pengembangan aplikasi. Adapun metode pengembangan perangkat lunak pada penelitian ini ialah Rational Unified Prosess (RUP).

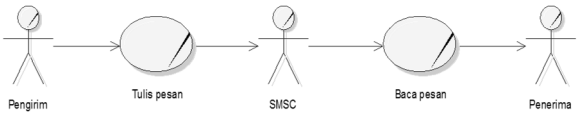


Gambar 2.4 Arsitektur *Rational Unified Process* (RUP) (10)

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem yang Berjalan

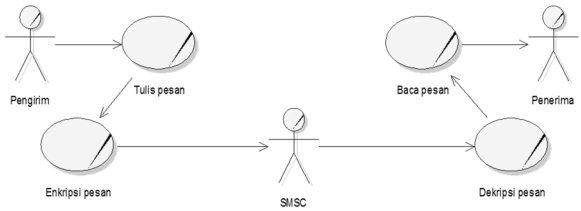
Analisis sistem yang berjalan pada penelitian ini pengguna/*user* belum mendapatkan keamanan dalam pertukaran informasi melalui *Short Message Service* (SMS) pada ponsel BlackBerry. Karena operator *Short Message Service Center* (SMSC) atau pihak yang lainnya dapat menyadap informasi dari pengirim pesan ke penerima pesan atau sebaliknya.



Gambar 3.1 Proses bisnis dari sistem yang sedang berjalan

3.2 Analisis Sistem yang Diajukan

Dari analisis sistem yang berjalan di atas dapat dilihat bahwa sistem tersebut dapat dikatakan tidak aman untuk digunakan karena pengguna/*user* tidak mendapatkan kerahasiaan informasi, untuk itu dibutuhkan suatu sistem yang baru yang dapat memberikan keamanan kepada pengguna/*user* dalam memberikan layanan pertukaran informasi. Sistem enkripsi dan dekripsi pesan dengan algoritma *Rivest Code* 6 (RC6) sangat tepat untuk diimplementasikan pada sistem yang akan dibangun di atas *platform* BlackBerry.



Gambar 3.2 Proses bisnis dari sistem yang diajukan

3.2.1 Analisis Karakteristik Pengguna/*User*

Perangkat lunak bantu ini dapat digunakan oleh siapapun, mulai dari anak-anak sampai dewasa jika pengguna menggunakan ponsel BlackBerry.

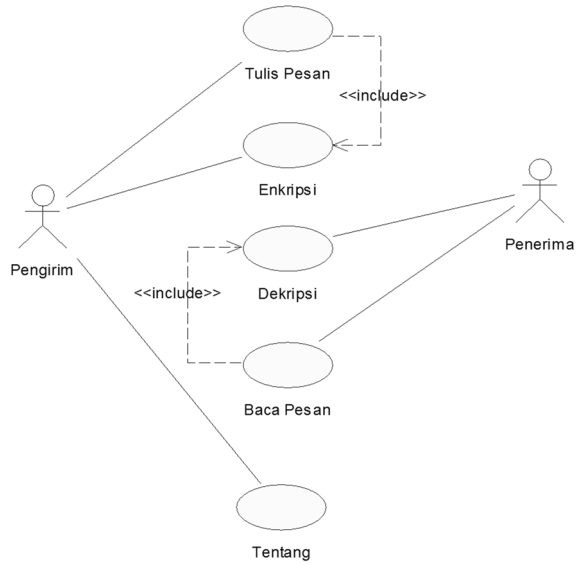
Pengguna/*user* dapat menggunakan perangkat lunak bantu ini dengan tujuan mengamankan pesan dengan media *Short Message Service* (SMS) untuk bertukar informasi kepada sesama pengguna ponsel BlackBerry. Adapun karakteristik pengguna/*user* dari perangkat lunak bantu ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

No	Nama Pengguna	Layanan
1	Pengirim	Dapat melakukan pengiriman pesan SMS dengan metode enkripsi.
2	Penerima	Dapat melakukan menerima pesan SMS dengan metode dekripsi.

3.2.2 Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan interaksi antara pengguna/*user* dengan sistem yang akan dibangun.



Gambar 3.4 Use Case Diagram Sistem yang diajukan.

Berikut ini merupakan penjelasan secara rinci dari use case diagram dalam bentuk tabel yang merupakan deskripsi dari use case diagram di atas yang ditunjukan pada Tabel 3.2.

Tabel 3.2 Deskripsi Use Case

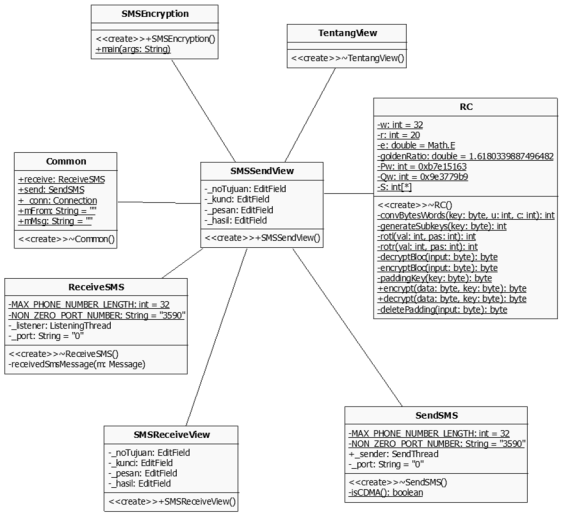
No	Aktor	Use Case	Keterangan
1	Pengirim	Tulis Pesan	Pengirim menulis pesan untuk aplikasi untuk dikirim.

Tabel 3.2 Deskripsi Use Case (Lanjutan)

2	Pengirim	Enkripsi Pesan	Pengirim mengenkripsi pesan yang akan dikirim agar pesan teracak dan pesan tidak dapat dibaca.
3	Penerima	Baca Pesan	Penerima mendapatkan pesan dari pengirim.
4	Penerima	Dekripsi Pesan	Penerima mendekripsi pesan yang teracak agar dapat membaca pesan.
5	Pengirim /Penerima	About	Pengirim/Penerima mendapatkan informasi dari aplikasi, seperti versi dan pembuat aplikasi.

3.2.3 Class Diagram

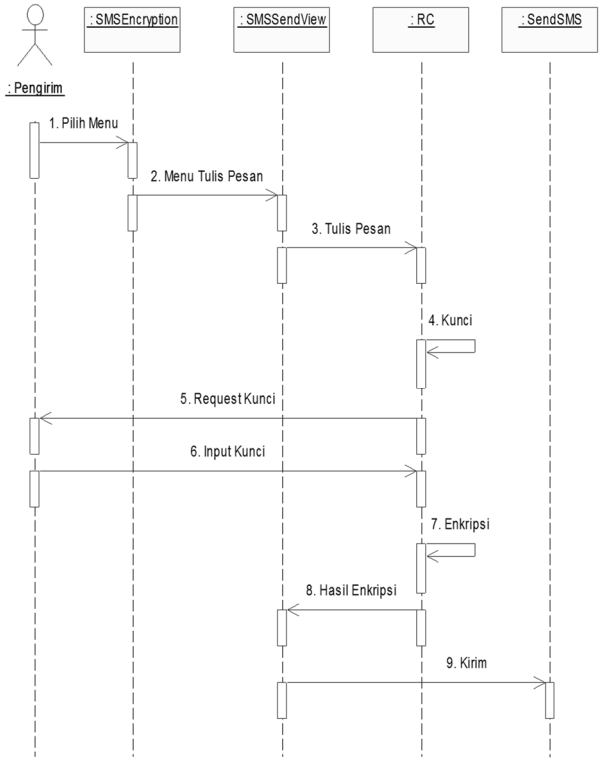
Class diagram merupakan diagram yang digunakan untuk menampilkan beberapa kelas paket yang ada dalam sistem/perangkat lunak yang akan dibangun.



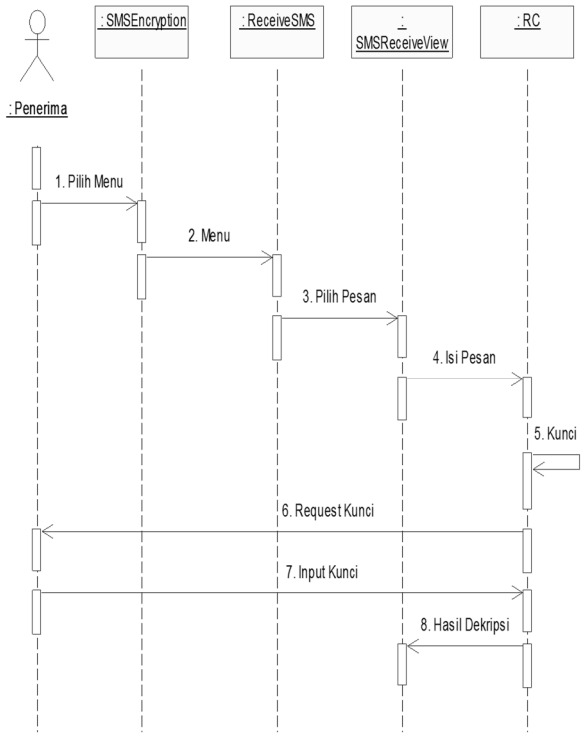
Gambar 3.5 Class Diagram RC6

3.2.4 Sequence Diagram

Sequence diagram merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan, pesan (message) apa yang dikirim dan kapan pelaksanaannya. Sequence diagram diatur berdasarkan waktu.



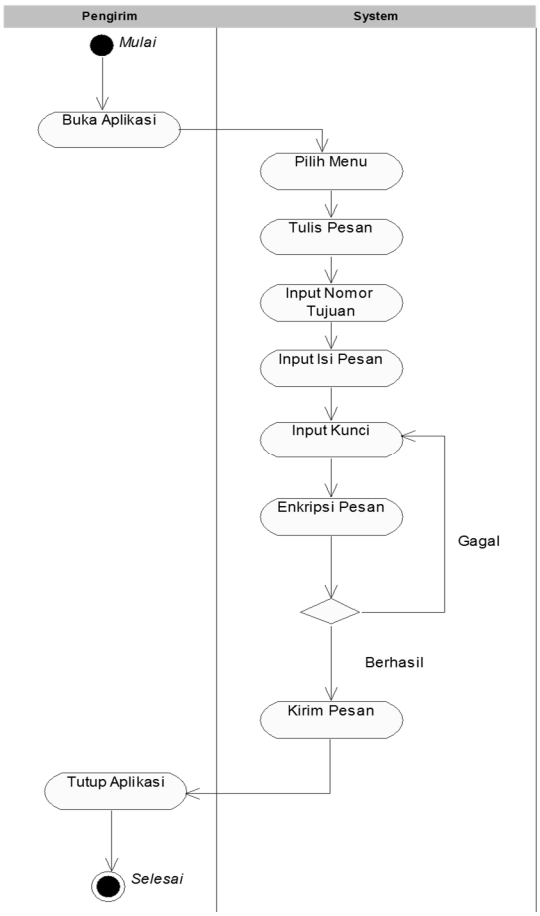
Gambar 3.6 Squence Diagram Pengirim Pesan



Gambar 3.7 Squence Diagram Penerima Pesan

3.2.5 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.



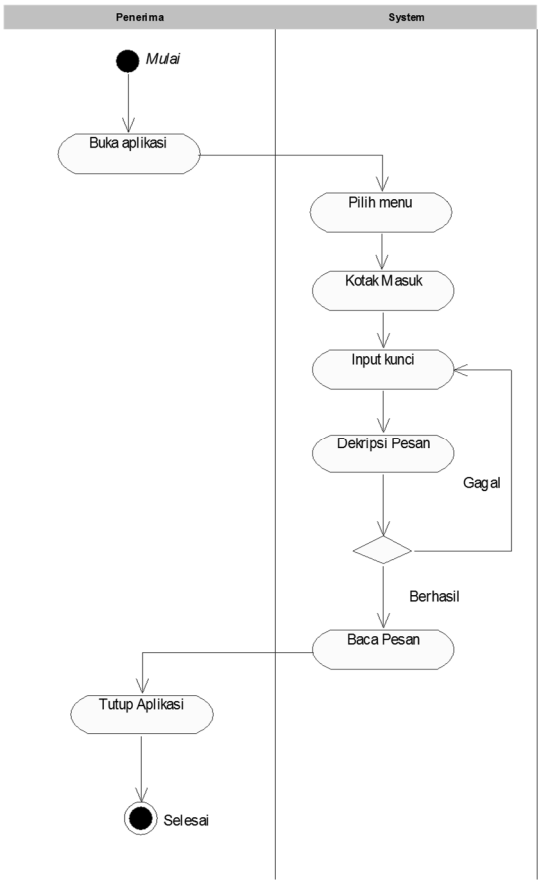
Gambar 3.8 Rancangan Diagram Aktivitas Pengiriman Pesan Dengan Teknik Enkripsi

Dari activity diagram di atas dapat dijelaskan pada skenario/prosedur dibawah ini :

1. Pengirim pesan membuka aplikasi kripto RC6 yang dibangun di atas ponsel BlackBerry.
2. Pengirim pesan akan diberikan pilihan menu oleh *system*.
3. Setelah pengirim pesan masuk ke menu, maka *system* akan memberikan pilihan kepada pengirim pesan untuk menulis pesan.
4. Setelah pengirim pesan selesai menulis isi pesan yang akan dikirim kepada penerima pesan, maka pengirim pesan disuguhkan *form* yang berisikan input kunci untuk melakukan enkripsi.
5. Setelah pengirim pesan melakukan *input* kunci, maka pengirim pesan disuguhkan sebuah *button*. *Button*

tersebut berisikan operasi untuk melakukan proses enkripsi.

6. Setelah proses *input* isi pesan SMS, input kunci, dan menekan *button* enkripsi selesai, maka pengirim pesan akan disuguhkan sebuah *form* untuk input nomor penerima pesan
7. Jika proses *input* isi pesan SMS, *input* kunci, menekan *button* enkripsi, input nomor penerima pesan dan menekan *button* pengiriman selesai, maka pengiriman pesan berhasil. Jika tidak berhasil maka *system* akan kembali mengulang menampilkan menu *input* isi pesan SMS, *input* kunci, menekan *button* enkripsi, input nomor penerima pesan.
8. Setelah langkah-langkah tersebut selesai sesuai dengan prosedur, maka pengirim pesan bisa menutup aplikasi.



Gambar 3. 9 Rancangan Diagram Aktivitas Penerimaan Pesan Dengan Teknik Dekripsi

Dari activity diagram di atas dapat dijelaskan pada skenario/prosedur dibawah ini :

1. Penerima pesan membuka aplikasi kripto RC6 yang dibangun di atas ponsel BlackBerry.
2. Penerima pesan akan diberikan pilihan menu oleh *system.*
3. Penerima pesan membuka *inbox* (kotak masuk)
4. Setelah membuka menu inbox, maka penerima pesan membuka pesan yang telah dikirm oleh pengirim pesan.
5. Karena pesan yang telah dikirim oleh pengirim pesan telah dienkripsi, maka diperlukan *input* kunci untuk membuka isi pesan tersebut.
6. Jika kunci sudah dimasukan oleh penerima pesan, maka proses selanjutnya ialah proses dekripsi pesan untuk melihat makna dari isi pesan tersebut.
7. Jika kunci yang dimasukan *valid* sesuai dengan kunci yang digunakan untuk enkripsi pesan, maka pesan berhasil didekripsi oleh penerima pesan. Jika tidak maka penerima pesan kembali untuk *input* kunci sampai kunci benar-benar *valid*.
8. Setelah langkah-langkah tersebut selesai sesuai dengan prosedur, maka pengirim pesan bisa menutup aplikasi.

4.1 Impelementasi Antarmuka Aplikasi

Adapun tampilan utama aplikasi ini merupakan menu utama yang diakses oleh pengguna sebelum masuk kepada halaman-halaman aplikasi yang akan digunakan.

Gambar 4.1 Tampilan Utama Aplikasi SMS Kripto

4.2 Tulis Pesan

Halaman ini merupakan tampilan dari aplikasi SMS Kripto. Halaman ini merupakan *core* dari aplikasi yang mempunyai fungsi untuk menulis pesan dan menirim pesan kepeda penerima pesan yang di dalamnya terdapat fungsi untuk enkripsi pesan dengan algoritma RC6.

Gambar 4.2 Tampilan Tulis Pesan Aplikasi SMS Kripto

4.3 Kotak Masuk

Halaman ini merupakan tampilan dari aplikasi SMS Kripto. Halaman ini merupakan *core* dari aplikasi yang mempunyai fungsi untuk membaca pesan dari pengirim pesan yang di dalamnya terdapat fungsi untuk dekripsi pesan dengan algoritma RC6.

Gambar 4.2 Tampilan Kotak Masuk dari Aplikasi SMS Kripto

4.3.1 Pengujian Enkripsi Pesan

Pengujian ini dilakukan untuk memeriksa apakah Aplikasi SMS Kripto tersebut berjalan sebagaimana mestinya, seperti *error handling*, kesesuaian hasil enkripsi pesan.

Gambar 4.3 Tampilan Pengujian *Input* dan Enkripsi Pesan

4.3.2 Pengujian Dekripsi Pesan

Pengujian ini dilakukan untuk memeriksa apakah Aplikasi SMS Kripto tersebut berjalan sebagaimana mestinya, seperti *error handling*, kesesuaian hasil dekripsi pesan.

Gambar 4.3 Tampilan Pengujian *Input* dan Enkripsi Pesan

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari implementasi Algoritma Rivest Code 6 (RC6) untuk enkripsi dan dekripsi pesan sms pada ponsel *BlackBerry* dapat diambil beberapa kesimpulan, yaitu :

1. Berhasil dalam rancang bangun perangkat lunak yang diimplementasikan di ponsel *BlackBerry* untuk mengamankan pesan SMS.
2. Behasil menerapkan algoritma Rivest Code 6 (RC 6) pada ponsel *BlackBerry*,

3. Berhasil dalam melakukan proses kriptografi, yaitu proses enkripsi dan dekripsi pesan SMS pada ponsel BlackBerry.
4. Jika dilihat dari segi performa, maka aplikasi SMS bawaan dari ponsel BlackBerry lebih cepat, dikarenakan aplikasi tersebut tidak melalukan proses enkripsi dan dekripsi pesan. Dan,
5. Jika dibandingkan dengan algoritma sebelum *Rivest Code* 6 (RC 6) yaitu algoritma *Rivest Code* 5 (RC 5), dan algoritma *Rivest Code* 4 (RC 4). Algoritma *Rivest Code* 6 (RC 6) memiliki kelebihan yaitu lebih cepat, sederhana, dan aman.

5.2 Saran

Penulis menyadari bahwa terdapat kekurangan dalam pengembangan Aplikasi Kripto dengan menggunakan algoritma *Rivest Code* 6 (RC6) pada *flatform BlackBerry*. Oleh karena itu, saran yang dapat diberikan yakni :

1. Tampilan *user interfaces* (UI) masih sederhana, akan lebih baik jika dikembangkan dengan *user interfaces* (UI) yang lebih menarik lagi.
2. Aplikasi Kripto ini hanya dapat dijalan pada sistem operasi Blackberry versi 7.1, akan lebih baik lagi jika dikembangkan ke Blackberry versi 10 yang sejajar dengan sistem operasi Android dan IOS.
3. Aplikasi ini memiliki panjang kunci 4 byte (4 karakter), akan lebih baik jika panjang kunci ditambahkan.
4. Aplikasi kripto ini hanya dapat melakukan proses kriptografi pada pesan teks, akan lebih baik jika dapat melakukan proses kriptografi pada pesan multimedia

DAFTAR PUSTAKA

Bishop, D. (2003). *Introduction to Cryptograhpy with JAVA™ Applets.* Canada: Jones and Bartlett Publishers.

Chandra, P., Messier, M., & Viega, J. (2002). *Network Security with OpenSSL.* United States: O'Reilly.

H, N. S. (2013). *Aplikasi Berbasis Android.* Bandung: Penerbit Informatika.

Katz, J., & Lindell, Y. (2008). *Introduction to Modern Cryptography.* New York: Chapman & Hall/CRC.

Larman, C. (2004). *Applying UML and Pattern. An Introduction to Object Oriented Analys and Design and the Unified Proses.* United States: Prentice Hall.

Levitin, A. (2003). *Introduction to The Design & Analysis of Algorithms 3rd Edition.* Boston: Addison-Wesley.

Munir, R. (2006). *Kriptografi.* Bandung: Informatika.

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography.* Berlin: Springer.

Permana, R. W. (2008). Implementasi Algoritma RC6 Untuk Enkripsi SMS Pada Telepon Selular. *Jurnal Digital*, 1-7.

Prayudi, Y., & Halik, I. (2005). Studi dan Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 1-10.

Pressman, R. S. (1992). *Software Engineering: A Practitioner's Aprroach.* Singapore: McGraw-Hill.

Rivest, R., Robshaw, M., Sidney, R., & Yin, Y. (1998). The RC6 Block Cipher. *AES Submission*, 1-21.

Schneier, B. (1996). *Applied Cryptography 2nd.* Boston: John Wiley & Sons.

Uswiratri, Y. (2011). Implementasi Algoritma RC6 Untuk Enkripsi Citra MMS Dengan Menggunakan J2ME. *Skripsi*, 1-20.

Wargo, J. M. (2009). *BlackBerry® Development Fundamentals.* Boston: Addison-Wesley.