Perancangan Perangkat Lunak Pembelajaran Kriptografi Metode Camellia

Erianto Ongko[1)] Andi Suwanto[2)]
STMIK IBBI Medan
Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548
Email: erianto_ongko@yahoo.co.id[1]

Abstrak

Metoda kriptografi ditujukan untuk mengamankan data / informasi yang bersifat rahasia agar tidak diketahui oleh pihak-pihak yang tidak berkepentingan. Dalam ilmu kriptografi, banyak metoda yang dapat digunakan untuk mengamankan data. Setiap metoda memiliki kelebihan dan kekurangannya masing-masing. Namun, yang menjadi penghalang utama dalam memilih metoda kriptografi yang cocok adalah bagaimana mengetahui dan memahami cara kerja algoritma dari metoda kriptografi tersebut.

Salah satu metode kriptografi yang ada adalah metode Camellia. Camellia dikembangkan oleh NTT (Nippon Telegraph and Telephone Corporation) dan Mitsubishi Electric Corporation pada tahun 2000. Camellia merupakan *block cipher* yang dirancang oleh ahli-ahli dalam riset dan pengembangan teknik kriptografik. Camellia mempunyai *interface* yang sama dengan *Advanced Encryption Standard* (AES). Algoritma enkripsi dan dekripsi Camellia diputar dengan menggunakan *secret key* yang panjang 128/192/256-bit dan dengan blok data sebesar 128-bit

Perangkat lunak pembelajaran ini mampu menampilkan tahapan-tahapan penyelesaian untuk proses pembentukan kunci, enkripsi dan dekripsi secara langkah demi langkah. Perangkat lunak pembelajaran juga menambahkan tampilan bagan algoritma dari metoda Camellia untuk membantu pemahaman terhadap cara kerja algoritma dari metoda Camellia.

Kata Kunci: Kriptografi, Enkripsi, dan Dekripsi

Abstract

Cryptographic methods aimed at securing data / confidential information that is not known by those who are not interested. In the science of cryptography, many methods that can be used to secure the data. Each method has advantages and disadvantages of each. However, that a major barrier in choosing a suitable method of cryptography is to know and understand the workings of the cryptographic algorithms of the method.

One method of cryptography that there is a method of Camellia. Camellia developed by NTT (Nippon Telegraph and Telephone Corporation) and Mitsubishi Electric Corporation in 2000. Camellia is a block cipher designed by experts in the research and development of cryptographic techniques. Camellia has the same interface with the Advanced Encryption Standard (AES). Camellia encryption and decryption algorithms rotated by using the secret key length 128/192/256-bit and the data blocks of 128-bit

Learning software is able to show the stages of completion of the process of the formation of the key, encryption and decryption are step by step. Learning software also adds the tree view algorithm of Camellia methods to aid understanding of how the algorithm of Camellia method.

***Keywords**: Cryptography, Encryption, and Decryption*

1. Pendahuluan

Salah satu masalah penting dalam kehidupan sehari-hari adalah masalah keamanan data yaitu mengamankan data dari orang yang tidak berhak (http://en.wikipedia.org/cryptography). Pengamanan data terutama diperlukan pada saat pengiriman data dari seseorang kepada orang lain karena data paling mudah dibajak pada saat proses pengiriman. Pengamanan data menjadi begitu penting sebab data tersebut dapat saja merupakan sesuatu yang vital atau pribadi bagi pemiliknya. Terdapat banyak cara untuk menerapkan pengamanan data ini, salah satunya adalah dengan menggunakan kriptografi. Fungsi kriptografi adalah membuat data walaupun berhasil dibajak menjadi tidak dapat dibaca oleh pembajaknya.

Seiring dengan peningkatan kepentingannya, banyak metode-metode yang ditemukan maupun diperluas penggunaannya. Jenis metode kriptografi terdiri atas metode kunci privat dan metode kunci publik. Contoh-contoh dari metode tersebut adalah metode Frog, Gost, RSA, RC5, RC6, Sapphire, TwoFish, BlowFish dan lain-lain. Diantara metode-metode tersebut terdapat metode yang hanya membutuhkan operasi matematika yang sederhana, tetapi juga terdapat metode yang melibatkan teori yang rumit dan sulit implementasinya. Khusus untuk pemanfaatan operasi matematika biasanya menggunakan operasi *bit* yang perhitungannya memerlukan fungsi iterasi dan operasi seperti *and*, *xor*, dan *or* yang diputar dalam setiap prosesnya. Proses operasi tersebut biasanya terdiri atas 3 (tiga) tahapan yaitu pembentukan kunci yaitu proses untuk menghasilkan kunci efektif berdasarkan kunci atau *password* yang dimasukkan oleh *user*. Dilanjutkan dengan proses enkripsi yaitu proses mengubah pesan yang dapat dibaca menjadi pesan dalam bentuk lain berdasarkan hasil pembentukan kunci. Sedangkan proses untuk membalikkan pesan yang telah terenkripsi menjadi bentuk pesan yang dapat dibaca disebut proses dekripsi.

Dalam pembelajaran mata kuliah kriptografi sering sekali mahasiswa kesulitan dalam memahami cara kerja serta proses perhitungan pada suatu metode kriptografi karena banyaknya metoda dalam kriptografi serta terbatasnya waktu. Untuk itu dibutuhkan media tambahan sebagai alat bantu sehingga lebih efektif dan mudah dipahami, di mana dalam hal ini perangkat lunak pembelajaran sangat membantu dalam memahami suatu metode kriptografi. Salah satu metode kriptografi adalah Camellia yang dikembangkan oleh NTT (*Nippon Telegraph and Telephone Corporation*) dan *Mitsubishi Electric Corporation*.

Disebabkan luasnya permasalahan yang ada, maka peneliti memberikan batasan masalah antara lain:

1. Metode yang digunakan adalah metode Camellia.
2. Kunci yang digunakan untuk algoritma Camellia ini adalah sebesar 128 *bit* dengan ukuran blok proses sebesar 128 *bit* juga.
3. Program dapat menampilkan seluruh perhitungan dari algoritma Camellia tetapi tidak dalam bentuk animasi.
4. Semua operasi perhitungan akan dilakukan dalam bentuk operasi 8-bit, 32-*bit* dan 64-*bit*.
5. Perangkat lunak tidak dapat mengenkripsi data.

2. Metode Penelitian

Pembelajaran adalah setiap perubahan perilaku yang relatif permanen, terjadi sebagai hasil dari pengalaman. Definisi sebelumnya menyatakan bahwa seorang manusia dapat melihat perubahan terjadi tetapi tidak pembelajaran itu sendiri. Konsep tersebut adalah teoretis, dan dengan demikian tidak secara langsung dapat diobservasi.

Perangkat Ajar yang dibantu oleh komputer atau *Computer Assisted Instruction* (CAI) adalah pengajaran dengan menggunakan perangkat aplikasi (*application software*) yang dirancang untuk menghasilkan metode dan materi pengajaran suatu topik dengan tujuan memberikan fasilitas belajar yang lebih mudah.

Definisi lain menyatakan CAI adalah suatu istilah yang mengacu ke situasi belajar dimana siswa berinteraksi dengan komputer dan juga dibimbing oleh komputer melalui suatu pelajaran yang bertujuan untuk mencapai tujuan tertentu.

Camellia dikembangkan oleh NTT (*Nippon Telegraph and Telephone Corporation*) dan *Mitsubishi Electric Corporation* pada tahun 2000. Camellia merupakan *block cipher* yang dirancang oleh ahli-ahli dalam riset dan pengembangan teknik kriptografik. Camellia mempunyai *interface* yang sama dengan *Advanced Encryption Standard* (AES). Camellia mempunyai tingkat efisiensi yang baik dan dapat diimplementasikan secara efisien dalam *software* pada banyak *platform* dan dapat juga diimplementasikan dalam *hardware* yang kompak dan mengkonsumsi daya yang rendah. Algoritma enkripsi dan dekripsi Camellia diputar dengan menggunakan *secret key* yang panjang 128/192/256-bit dan dengan blok data sebesar 128-bit (Aoki, 2001: 1)

Camellia dikembangkan dalam suatu tim yang terdiri atas 7 orang yaitu: Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, dan Toshio Tokita seperti ditunjukkan pada Gambar 1.

Gambar 1. Tim Pembuat Camelia

2.1. Model

Pada diagaram di bawah proses awal adalah *user* menentukan *key*, *plaintext* atau *ciphertext*, serta *delay* waktu proses dari program. Langkah pertama adalah memverifikasi data inputan dengan tujuan apakah data yang diinput sudah benar atau tidak kurang. Selanjutnya pada tahap kedua akan dilakukan proses pembentukan kunci dengan menggunakan algoritma pembentukan Camellia. Hasil yang didapat merupakan sub kunci yang dapat digunakan untuk proses enkripsi atau dekripsi. Tahapan selanjutnya melakukan proses enkripsi/dekripsi sesuai pilihan yang ditentukan oleh *user*. Setiap langkah baik dalam proses pembentukan kunci dan proses enkripsi/dekripsi akan ditampilkan dalam layar penampil program dimana sebelumnya diformat terlebih dahulu. Hasil akhir dari proses perhitungan algoritma Camellia dapat ditampilkan sebagai media pembelajaran terhadap algoritma ini seperti ditunjukkan pada Gambar 2.

Gambar 2. Alur Cara Kerja Program

Agar proses ataupun cara kerja dari algoritma Camellia dapat lebih dipahami maka pada bagian berikut ini peneliti akan menyertakan sebuah contoh kasus dengan input panjang kunci sebesar 128 *bit* dan *ciphertext* sebesar 128 *bit* juga. Perhitungan yang dilakukan mencakup pembentukan kunci, perhitungan fungsi F, FL, dan FLINV, serta proses enkripsi dan proses dekripsi.

2.2. Analisis

Secara mendasar Camellia dapat dibagi menjadi dua bagian besar yaitu bagian "*key scheduling*" dan bagian "*data randomizing*". (Aoki, et.al, 2000: 23-30)

Untuk menjelaskan cara kerja dari algoritma Camellia maka digunakan beberapa notasi dan terminologi untuk menyatakan operator yang digunakan dalam pembahasan ini agar penjelasan algoritma lebih dapat dipahami.

& operasi bitwise AND.
| operasi bitwise OR.
^ operasi bitwise exclusive-OR.
<< operasi logikal shift kiri.
>> operasi logikal shift kanan.
<<< operasi rotasi kiri.
~y bitwise complement dari y.

0x representasi hexadecimal.

Sebagai catatan operasi logikal *shift* kiri dilakukan dengan lebar data yang tidak terbatas. Konstanta nilai dari MASK8 (nilai konstanta untuk operasi 8 *bit*), MASK32 (nilai konstanta untuk operasi 32 *bit*), MASK64 (nilai konstanta untuk operasi 64 *bit*), dan MASK128 (nilai konstanta untuk operasi 128 *bit*) dinyatakan sebagai berikut:

```
MASK8   = 0xff;
MASK32  = 0xffffffff;
MASK64  = 0xffffffffffffffff;
MASK128 = 0xffffffffffffffffffffffffffffffff;
```

2.2.1. *Key Scheduling*

Pada bagian *key schedule* Camellia, variabel 128-bit KL dan KR didefinisikan sebagai berikut. Untuk kunci 128-bit, maka kunci K 128-bit digunakan sebagai KL dan KR merupakan nilai 0 (nol). Untuk kunci 192-bit, bagian paling kiri dari 128-bit dari kunci K digunakan sebagai KL dan pemotongan dari bagian paling kanan 64-bit dari K dan komplemen dari bagian paling kanan dari 64-bit K digunakan sebagai KR. Untuk kunci dengan panjang 256-bit, bagian paling kiri dari kunci K digunakan sebagai KL dan bagian bit paling kanan dari K digunakan sebagai KR.

```
128-bit key K:
    KL = K;    KR = 0;
  192-bit key K:
    KL = K >> 64;
    KR = ((K & MASK64) << 64) | (~(K & MASK64));
  256-bit key K:
    KL = K >> 128;
    KR = K & MASK128;
```

Variabel KA dan KB dengan panjang 128-bit digenerasi dari KL dan KR dengan aturan berikut. Sebagai catatan KB digunakan hanya jika panjang dari kunci rahasia (*secret key*) mempunyai panjang 192 atau 256 bit. D1 dan D2 merupakan variabel *temporary* dengan panjang 64-bit. Fungsi F akan dijelaskan pada sub bab mengenai fungsi F.

```
D1 = (KL ^ KR) >> 64;
D2 = (KL ^ KR) & MASK64;
D2 = D2 ^ F(D1, Sigma1);
D1 = D1 ^ F(D2, Sigma2);
D1 = D1 ^ (KL >> 64);
D2 = D2 ^ (KL & MASK64);
D2 = D2 ^ F(D1, Sigma3);
D1 = D1 ^ F(D2, Sigma4);
KA = (D1 << 64) | D2;
D1 = (KA ^ KR) >> 64;
D2 = (KA ^ KR) & MASK64;
D2 = D2 ^ F(D1, Sigma5);
D1 = D1 ^ F(D2, Sigma6);
KB = (D1 << 64) | D2;
```

Konstanta 64-bit yaitu Sigma1, Sigma2, ..., Sigma6 digunakan sebagai "*keys*" dalam fungsi F. Nilai konstanta tersebut dalam notasi heksadesimal adalah sebagai berikut:

```
Sigma1 = 0xA09E667F3BCC908B;
Sigma2 = 0xB67AE8584CAA73B2;
Sigma3 = 0xC6EF372FE94F82BE;
Sigma4 = 0x54FF53A5F1D36F1C;
Sigma5 = 0x10E527FADE682D1D;
Sigma6 = 0xB05688C2B3E6C1FD;
```

Sub key 64-bit digenerasi dengan merotasikan KL, KR, KA, dan KB dan mengambil setengah bagian kiri dan kanan dari hasil tersebut.

Untuk kunci 128-bit, *sub key* 64-bit yaitu kw1, ..., kw4, k1, ..., k18, ke1, ..., ke4 digenerasi sebagai berikut:

```
kw1 = (KL <<<   0) >> 64;
kw2 = (KL <<<   0) & MASK64;
```

k1 = (KA <<< 0) >> 64;
k2 = (KA <<< 0) & MASK64;
k3 = (KL <<< 15) >> 64;
k4 = (KL <<< 15) & MASK64;
k5 = (KA <<< 15) >> 64;
k6 = (KA <<< 15) & MASK64;
ke1 = (KA <<< 30) >> 64;
ke2 = (KA <<< 30) & MASK64;
k7 = (KL <<< 45) >> 64;
k8 = (KL <<< 45) & MASK64;
k9 = (KA <<< 45) >> 64;
k10 = (KL <<< 60) & MASK64;
k11 = (KA <<< 60) >> 64;
k12 = (KA <<< 60) & MASK64;
ke3 = (KL <<< 77) >> 64;
ke4 = (KL <<< 77) & MASK64;
k13 = (KL <<< 94) >> 64;
k14 = (KL <<< 94) & MASK64;
k15 = (KA <<< 94) >> 64;
k16 = (KA <<< 94) & MASK64;
k17 = (KL <<< 111) >> 64;
k18 = (KL <<< 111) & MASK64;
kw3 = (KA <<< 111) >> 64;
kw4 = (KA <<< 111) & MASK64;

Untuk kunci 192-bit dan 256-bit, maka sub key 64-bit sub keys kw1, ..., kw4, k1, ...,k24, ke1, ..., ke6 akan digenerasi dengan cara sebagai berikut:

kw1 = (KL <<< 0) >> 64;
kw2 = (KL <<< 0) & MASK64;
k1 = (KB <<< 0) >> 64;
k2 = (KB <<< 0) & MASK64;
k3 = (KR <<< 15) >> 64;
k4 = (KR <<< 15) & MASK64;
k5 = (KA <<< 15) >> 64;
k6 = (KA <<< 15) & MASK64;
ke1 = (KR <<< 30) >> 64;
ke2 = (KR <<< 30) & MASK64;
k7 = (KB <<< 30) >> 64;
k8 = (KB <<< 30) & MASK64;
k9 = (KL <<< 45) >> 64;
k10 = (KL <<< 45) & MASK64;
k11 = (KA <<< 45) >> 64;

k12 = (KA <<< 45) & MASK64;
ke3 = (KL <<< 60) >> 64;
ke4 = (KL <<< 60) & MASK64;
k13 = (KR <<< 60) >> 64;
k14 = (KR <<< 60) & MASK64;
k15 = (KB <<< 60) >> 64;
k16 = (KB <<< 60) & MASK64;
k17 = (KL <<< 77) >> 64;
k18 = (KL <<< 77) & MASK64;
ke5 = (KA <<< 77) >> 64;
ke6 = (KA <<< 77) & MASK64;
k19 = (KR <<< 94) >> 64;
k20 = (KR <<< 94) & MASK64;
k21 = (KA <<< 94) >> 64;
k22 = (KA <<< 94) & MASK64;
k23 = (KL <<< 111) >> 64;

```
k24 = (KL <<< 111) & MASK64;
kw3 = (KB <<< 111) >> 64;
kw4 = (KB <<< 111) & MASK64;
```

2.3. Desain dan Implementasi

Perangkat lunak pembelajaran kriptografi metoda Camellia ini dapat dijalankan cara sebagai berikut :

1. Lakukan proses instalasi dan ikuti petunjuk pada layar.
2. Jalankan menu Start → Program → Camellia ataupun melalui klik pada file Camellia.EXE.

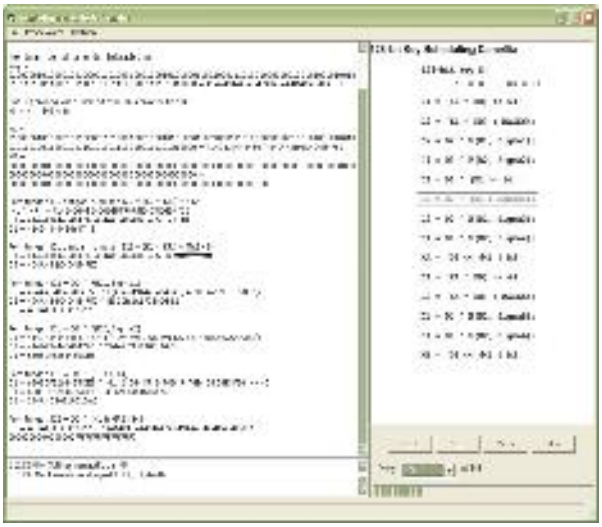
Setelah itu layar akan ditampilkan sebuah tampilan menu seperti terlihat pada Gambar 3 berikut ini.

Gambar 3 Tampilan Menu

Pada tampilan menu akan dijelaskan secara singkat mengenai Camellia dan penjelasan lainnya seperti jenis operator, kunci, *block size*, generasi kunci, enkripsi, dan dekripsi. *User* dapat menampilkan keterangan singkat dengan mengklik pada label di bagian menu. Untuk memulai pembelajaran maka *user* dapat mengklik pada bagian label pada *flow chart* di sebelah kanan. Urutan pembelajaran yang dilakukan dapat berpatokan pada bentuk *flow chart* ini.

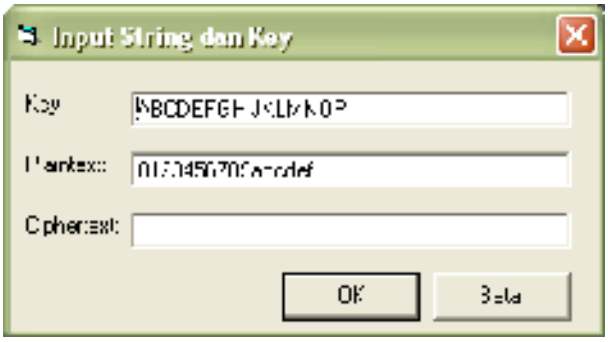
Pada tampilan menu ini terdapat 3 (tiga) buah *button* yaitu ”Proses” digunakan untuk menampilkan *form* pembelajaran utama, *button* ”Teori” untuk menampilkan seluruh materi dari Camellia dalam bentuk HTML, dan *button* ”Keluar” untuk keluar dari aplikasi ini.

Untuk itu jika *label* ”Kunci” diklik maka akan ditampilkan sebuah tampilan *form* seperti ditunjukkan pada Gambar 4 berikut ini:



Gambar 4 Tampilan *Form* Generasi Kunci

Pada tampilan ini langkah pertama yang harus dilakukan adalah meng-*input* parameter seperti terlihat pada Gambar 5 berikut ini.



Gambar 5 Tampilan *Input Parameter*

Isikan pada *field* ”*Key*” dan ”*Plaintext*” sesuain dengan ukurannya yaitu 128-bit atau 16 karakter. Setelah itu klik pada tombol ”OK” dan tombol ”Batal” untuk pembatalan dan kembali pada *form* sebelumnya.

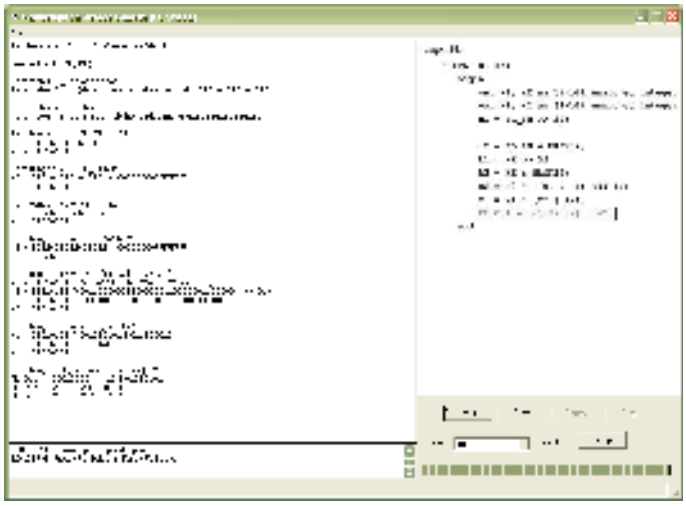
Kemudian tentukan kecepatan proses di bagian *delay* yang berupa *combo box*. Setelah itu klik pada tombol ”*Start*” untuk memulai proses generasi kunci. Gunakan tombol ”*Pause*” untuk menghentikan proses sejenak dan tombol ”*Stop*” untuk menghentikan proses. Status kemajuan proses dapat dilihat pada bagian *progress bar*.

Bagian berikutnya adalah pembelajaran fungsi F (Gambar 5). Seperti halnya dengan pembelajaran generasi kunci maka langkah pertama adalah meng-*input* parameter terlebih dahulu dengan mengklik pada tombol ”*Input*” dan tampilan kotak input akan dimunculkan seperti terlihat pada Gambar 6, Gambar 7 dan Gambar 8 menunjukkan perhitungan fungsi FL dan fungsi FL Invers.

Gambar 5 Tampilan Pembelajaran Fungsi F

Semua tombol yang ada penjelasannya sama seperti kegunaan tombol pada tampilan generasi kunci. Untuk menutup *form* ini *user* dapat mengklik pada tombol ”Tutup”.

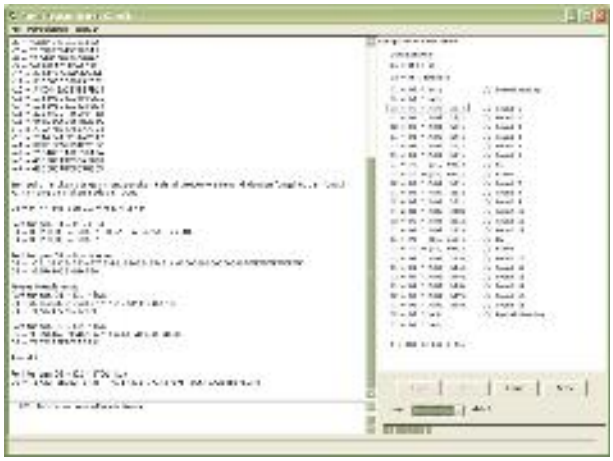
Gambar 6 Tampilan *Input* Parameter Fungsi



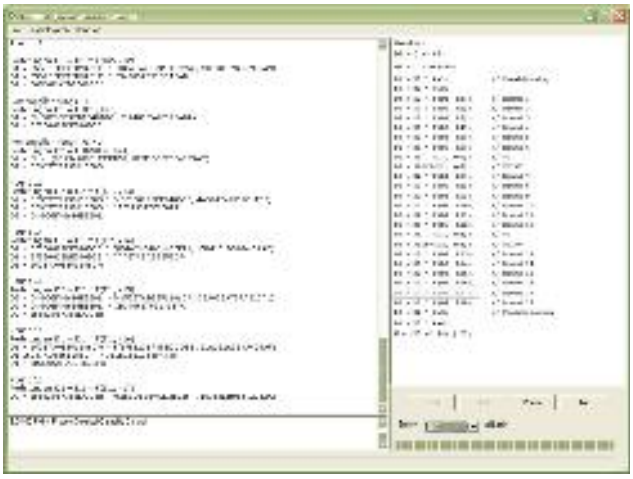
Gambar 7 Tampilan Pembelajaran Fungsi FL

Gambar 8 Tampilan Pembelajaran Fungsi FL Invers

Selanjutnya adalah tampilan proses enkripsi dan dekripsi (Gambar 9 dan Gambar 10). Pada tampilan ini digunakan untuk menampilkan rangkaian proses perhitungan algoritma Camellia untuk panjang kunci dan *block size* masing-masing 128-bit. Semua tombol yang ada penjelasannya sama seperti kegunaan tombol pada tampilan generasi kunci.



Gambar 9 Tampilan Proses Enkripsi



Gambar 10 Tampilan Proses Dekripsi

4. Kesimpulan dan Saran

4.1. Kesimpulan

peneliti menarik kesimpulan sebagai berikut :

1. Program yang dirancang dapat menampilkan alur proses seluruh diagram baik untuk proses pembentukan kunci, enkripsi, serta dekripsi Camellia sesuai dengan pembahasan yang dilakukan pada bab-bab sebelumnya.
2. Dengan adanya perangkat lunak pembelajaran kriptografi metode Camellia maka cara kerja metode Camellia dapat dipahami secara langkah per langkah.
3. Dengan adanya perangkat lunak pembelajaran ini maka dapat digunakan sebagai alat bantu untuk mempelajari algoritma Camellia dengan fasilitas menggunakan tampilan berupa animasi algoritma yang bergerak sejalan dengan proses perhitungan.
4. Camellia merupakan salah satu bentuk kriptografi *block cipher* yang mempunyai panjang kunci variasi dari 128 bit, 192 bit dan 256 bit dengan ukuran blok yang diproses sebesar 128 bit dimana pada implementasi perangkat lunak pembelajaran kriptografi metode Camellia menggunakan panjang kunci 128 bit.
5. Perangkat lunak pembelajaran kriptografi metode Camellia yang dirancang meliputi pembelajaran mengenai proses pembentukan kunci, enkripsi, dekripsi, fungsi F, fungsi FL, dan fungsi FL Invers.
6. Dengan adanya perancangan perangkat lunak pembelajaran ini, maka dapat dikatakan tujuan dari penelitian ini tercapai.

4.2. Saran

Peneliti memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak pembelajaran metoda kriptografi yaitu :

1. Dapat dipertimbangkan untuk menambahkan tutorial pada perangkat lunak pembelajaran agar lebih mudah dipahami.
2. Perangkat lunak pembelajaran ini dapat dikembangkan untuk menampilkan proses enkripsi dan dekripsi untuk *file*.
3. Untuk pengembangan lebih lanjut, perangkat lunak pembelajaran ini sebaliknya ditambahkan dengan fasilitas narasi yang akan menerangkan semua proses yang berlangsung.
4. Untuk menambah bobot pembelajaran, sebaiknya program pembelajaran ini dapat ditambahkan dengan fasilitas multimedia.
5. Menggabungkan metode ini dengan metode yang lain seperti untuk proses kunci dapat dengan metode kunci publik seperti RSA, El Gamal, atau Diffie-Hellman, proses enkripsi dilanjutkan dengan metode kriptografi lain.
6. Ditambahkan juga modul simulasi pola penyerangan terhadap algoritma Camellia.

Daftar Pustaka

Ariyus, Dony, **Kriptografi: Keamanan Data dan Komunikasi**, Penerbit Graha Ilmu, Yogyakarta, 2006.

Aoki, Kazumaro, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita, **Specification of Camellia-a 128-bit Block Cipher, Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation**, version 1.0: July 12, 2000, Version 2.0: September 26, 2001.

Ariyus, Dony, **Pengantar Ilmu Kriptografi: Teori, Analisis dan Implementasi**, Penerbit Andi, Yogyakarta, 2008.

Block Cipher Mode, http://en.wikipedia.org/wiki/Block_cipher_mode, tanggal akses 22 Juli 2010.

CAI, http://en.wikipedia.or/CAI, tanggal akses 22 Juli 2010.

Cryptography, http://en.wikipedia.org/cryptography, tanggal akses 22 Juli 2010.

Feistel Cipher, http://en.wikipedia.org/wiki/Feistel_cipher, tanggal akses 22 Juli 2010.

Fischer, Matthew, **How to implement the Data Encryption Standard (DES) - A step by step tutorial**, http://www.zone-h.org/download/file=770/, tanggal akses 15 Mei 2004.

Johnson, Michael Paul, **Beyond DES: Data compression and the MPJ encryption algorithm**, http://www.zone-h.org/download/file=804/, tanggal akses 15 Mei 2004.

Leman, **Metodologi Pengembangan Sistem**, PT. Elex Media Komputindo, Jakarta, 1998.

Mahyuzir, Tavri D., 1991, **Pengantar Analisis dan Perancangan Perangkat Lunak**, Penerbit Elex Media Komputindo, Jakarta.

Munir, Rinaldi, **Kriptografi**, Penerbit Informatika, 2006.

Murach, Mike & Associate, Inc, 1999, **Murach Visual Basic 6**, Tech Publications Pte Ltd, Singapore.

Pressman, Roger S., 1997, **Rekayasa Perangkat Lunak Pendekatan Praktisi**, Buku I, McGraw-Hill Book Co, Penerbit Andi, Yogyakarta.

Scheneier, Bruce, 1996, **Applied Cryptography**, Second Edition, John Wiley & Sons, Inc, Canada.

Sudirman, I., 2003, **Perkembangan Software Komputer**, Kuliah Pengantar Ilmu Komputer. http://www.ilmukomputer.com.

Wiener, Michael J., **Efficient DES Key Search**, http://www.zone-h.org/download/file=773/, tanggal akses 15 Mei 2010.