Simulasi Pengamanan File Teks Menggunakan Algoritma Massey-Omura

[1]Muhammad Reza, [1]Muhammad Andri Budiman, [1]Dedy Arisandi

[1]Program Studi S1 Teknologi Informasi
Fakultas Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara
E-mail: muhammadreza0101@yahoo.com, mandrib@gmail.com , dedyarisandi@usu.ac.id

Abstrak—Pada umumnya, kriptografi berhubungan dengan aktivitas menjaga komunikasi tetap rahasia dan khusus. *Encryption* (Enkripsi) adalah transformasi data ke dalam bentuk yang hampir tidak dapat dibaca tanpa pengetahuan yang cukup (misalnya kunci). *Decryption* (dekripsi) adalah kebalikan dari enkripsi, yaitu transformasi dari data yang telah dienkripsi (*ciphertext*) kembali ke bentuk semula (*plaintext*). Algoritma *Massey-Omura* merupakan sebuah pengembangan atas *Three-Pass Protocol* dan termasuk dalam kategori algoritma asimetris untuk kriptografi kunci publik. Sekuritas dari algoritma *Massey-Omura* ini terletak pada kesulitan menghitung logaritma diskrit dalam bidang terbatas sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sulit. Algoritma *Massey-Omura* ini juga membutuhkan metode *The Sieve of Eratosthenes* untuk membantu membangkitkan bilangan prima, algoritma *Euclidean GCD* untuk mencari dua buah bilangan bulat yang relatif prima dan untuk proses enkripsi menggunakan algoritma *Modulo Exponential* yang berfungsi menghitung nilai perpangkatan modulo bilangan besar serta algoritma *Modulo Invers* untuk proses dekripsi. Perangkat lunak ini aplikasinya dapat digunakan untuk mengenkripsi dan mendekripsi sebagai bentuk pengamanan *file* teks.

Kata kunci—*Kriptografi, Encryption, Decryption, Three-Pass Protocol, Massey-Omura , The Sieve of Eratosthenes, Euclidean GCD*.

I. PENDAHULUAN

Kemajuan dan perkembangan teknologi saat ini telah berpengaruh pada hampir semua aspek kehidupan manusia, tak terkecuali dalam hal berkomunikasi. Media komunikasi umum yang dapat digunakan oleh siapapun saat ini sangat rawan terhadap penyadapan informasi oleh pihak-pihak yang tidak berhak mengetahui informasi tersebut. Berbagai hal telah dilakukan untuk mendapatkan jaminan keamanan informasi rahasia ini. Salah satu cara yang bisa digunakan adalah menyandikan (mengenkripsi) informasi atau pesan rahasia yang akan dikirim, sehingga walaupun pihak yang tidak berkepentingan dapat membaca informasi tersebut, pihak tersebut tetap sulit untuk dapat memahami isi informasi tersebut tanpa menggunakan sebuah media tertentu [1].

Salah satu teknik untuk menjamin kerahasiaan informasi adalah dengan menggunakan teknik kriptografi. Informasi ini terlindung karena pesan asli akan diubah menjadi pesan *cipher* (pesan sandi) dengan menggunakan kunci tertentu sehingga pesan ini tidak dapat diketahui pihak yang tidak berkepentingan.

Algoritma *Massey-Omura* merupakan sebuah pengembangan atas *Three-Pass Protocol* dan termasuk dalam kategori algoritma asimetris untuk kriptografi kunci publik. Sekuritas dari algoritma *Massey-Omura* ini terletak pada kesulitan menghitung logaritma diskrit dalam bidang terbatas sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sulit.

Dalam kriptografi, *Three-Pass Protocol* dalam pengiriman pesan merupakan suatu kerangka kerja yang memungkinkan satu pihak untuk aman mengirim pesan ke pihak kedua tanpa perlu untuk bertukar atau mendistribusikan kunci enkripsi.

Disebut dengan *Three-Pass Protocol* karena pengirim dan penerima pesan melakukan pertukaran sebanyak tiga tahap untuk mengenkripsi pesan tersebut. *Three-Pass Protocol* pertama kali dikembangkan oleh Adi Shamir pada sekitar tahun 1980. Konsep dasar *Three-Pass Protocol* adalah bahwa masing-masing pihak memiliki kunci enkripsi pribadi dan sebuah kunci dekripsi pribadi. Kedua belah pihak menggunakan kunci mereka masing-masing untuk mengenkripsi dan untuk mendekripsi pesan.

Massey-Omura Cryptosystem diusulkan oleh James Massey dan Jim K. Omura pada 1982 sebagai pengembangan atas Algoritma *Three-Pass Protocol*. Cara kerja dari Algoritma *Massey-Omura* yaitu semua pengguna telah mensepakati kelompok batasan atas bidang tetap batasan *Fp* dengan *p* sebagai kekuatan utama. Setiap pengguna secara rahasia memilih acak bilangan bulat seperti :

$$0 \leq e < p - 1 \quad (1)$$

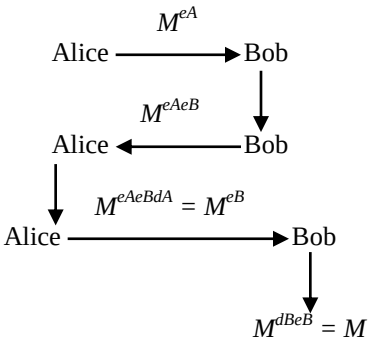
menghitung

$$GCD\ (e, p - 1) = 1 \quad (2)$$

dan menghitung

$$d = e^{-1} \bmod (p - 1) \quad (3)$$

Sebagai ilustrasi anggaplah bahwa Alice ingin mengirim pesan *M* yang aman untuk Bob, kemudian mereka mengikuti prosedur dimana :



II. IDENTIFIKASI MASALAH

Sistem yang akan dibangun untuk mengenkripsi dan mendekripsi pesan teks menggunakan algoritma *Massey-Omura*. Sekuritas dari algoritma *Massey-Omura* ini terletak pada kesulitan menghitung logaritma diskrit dalam bidang terbatas sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sulit. Maka dapat dibuat rumusan masalah yaitu : Bagaimana cara untuk menjaga keamanan *file*.

Mencakup batasan masalah yang akan diteliti adalah algoritma enkripsi dan dekripsi yang digunakan adalah algoritma *Massey-Omura*, Perangkat lunak yang digunakan untuk pengembangan sistem adalah Microsoft Visual Studio 2010, Pesan yang dapat dienkripsi dan didekripsi hanya berupa *file* (.txt), Pengujian sistem dilakukan dalam bentuk simulasi.

Dari permasalahan diatas, maka tujuan yang harus dicapai dan dilakukan dalam penelitian ini adalah sebagai berikut : Bagaimana untuk membuat sistem yang dapat menjaga keamanan *file* dan mengimplementasikan Algoritma *Massey-Omura* untuk enkripsi dan dekripsi dalam pembuatan sistem.

Manfaat yang diharapkan dari penelitian ini adalah hasil dari penelitian ini, aplikasinya dapat digunakan untuk mengenkripsi dan mendekripsi sebagai bentuk pengaman dalam *file* dan mendapatkan wawasan baru tentang kriptografi khususnya dalam pengamanan *file* menggunakan Algoritma *Massey-Omura*.

III. PENELITIAN TERDAHULU

Beberapa penelitian yang telah dilakukan pada sistem *Massey-Omura* yaitu meningkatkan (EMO-1) dengan menggantikan *modulo exponential* yang merupakan bagian dari dua bilangan prima yang besar. Dengan cara ini sistem dilengkapi dengan tingkat keamanan yang mirip dengan sistem kunci publik *RSA* dan (EMO-2) dengan menambahkan *digital signature* pada sistem EMO-1. *Digital signature* memungkinkan penerima pesan mengenkripsi dengan protokol EMO-2 untuk mengotentikasi identitas pengirim, menyediakan tambahan aspek keamanan [2].

IV. METODE PENELITIAN

A. Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Cryptos* berarti rahasia dan *graphia* berarti tulisan. Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga pesan ketika pesan dikirim dari suatu tempat ke tempat lain [3].

Secara umum, pengertian kriptografi adalah teknik yang digunakan untuk mengubah teks asli (*plaintext*) dengan menggunakan suatu kunci tertentu menjadi sebuah kode-kode yang tidak dimengerti (*chipertext*). Proses ini lebih dikenal dengan istilah enkripsi. Selanjutnya, *chipertext* yang ada dapat diubah menjadi teks asli semula dengan metode yang dikenal dengan istilah dekripsi.

Kriptografi klasik pada umumnya dienkripsi per karakter (menggunakan alfabet tradisional), sedangkan kriptografi modern beroperasi pada *string* dan biner. *Cipher* yang lebih kompleks seperti *RSA* dan *ElGamal* adalah algoritma modern yang sangat dikenal di dunia

kriptografi. Kriptografi modern tidak hanya berkaitan dengan teknik menjaga kerahasiaan pesan, tetapi juga menghasilkan tanda tangan digital dan sertifikat digital. Dengan kata lain, kriptografi modern tidak hanya memberikan aspek keamanan tetapi juga aspek-aspek lain yang dibutuhkan pada sistem keamanan informasi [4].

Adapun empat tujuan mendasar dari kriptografi adalah:

1. *Confidentiality*, adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. *Data integrity*, adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.
3. *Authentication*, adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication or entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*data origin authentication*).
4. *Non-repudiation*, adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengiriman pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

B. *The Sieve of Eratosthenes*

The Sieve of Eratosthenes merupakan sebuah algoritma klasik untuk menentukan seluruh bilangan prima sampai bilangan N yang ditentukan. Cara kerja dari metode ini adalah dengan melakukan eliminasi bilangan yang bukan bilangan prima untuk menyaring suatu kumpulan bilangan menjadi kumpulan bilangan prima [5].

Langkah-langkah penggunaan metode *the sieve of eratosthenes* dapat diuraikan sebagai berikut:

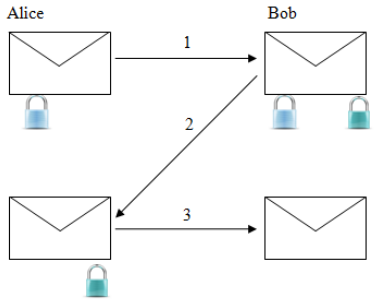
1. Pertama-tama, tuliskan daftar bilangan dari 2 sampai batas atas bilangan yang akan dicari.
2. Kemudian, tandai bilangan di dalam daftar yang merupakan kelipatan 2, dengan membiarkan bilangan 2 tetap tidak ditandai.
3. Lanjutkan ke bilangan berikutnya (dalam tahap ini adalah bilangan 3), dan tandai setiap kelipatan 3, dengan tetap membiarkan bilangan 3 tidak ditandai.
4. Lanjutkan ke bilangan berikutnya. Bila bilangan berikut tersebut telah ditandai, lanjutkan ke bilangan berikutnya.
5. Lanjutkan langkah penandaan seperti di atas sampai batas atas bilangan yang ditentukan.

C. *Three-Pass Protocol*

Dalam kriptografi, *Three-Pass Protocol* dalam pengiriman pesan merupakan suatu kerangka kerja yang memungkinkan satu pihak untuk aman mengirim pesan ke pihak kedua tanpa perlu untuk bertukar atau mendistribusikan kunci enkripsi.

Disebut dengan *Three-Pass Protocol* karena pengirim dan penerima pesan melakukan pertukaran sebanyak tiga tahap untuk mengenkripsi pesan tersebut. *Three-Pass Protocol* pertama kali dikembangkan oleh Adi Shamir pada sekitar tahun 1980. Konsep dasar *Three-Pass Protocol* adalah bahwa masing-masing pihak memiliki kunci enkripsi pribadi dan sebuah kunci dekripsi pribadi. Kedua belah pihak menggunakan kunci mereka masing-masing untuk mengenkripsi pesan dan kemudian untuk mendekripsi pesan.

Berikut merupakan Skema dari *Three-Pass Protocol* :



Gambar 1 Skema *Three-Pass Protocol*

Cara kerja skema *Three-Pass Protocol* [6]:

1. Pengirim (Alice) memilih sebuah kunci sandi pribadi *s* dan kunci dekripsi *t*. Pengirim pesan mengenkripsi pesan *m* dengan kunci *s* dan mengirimkan pesan terenkripsi *E(s, m)* untuk penerima (Bob)
2. Penerima memlilih sebuah kunci pribadi *r* dan kunci dekripsi *q* dan mengenkripsi pesan pertama *E(s, m)* dengan kunci *r* lalu, mengirimkan kembali kunci enkripsi ganda *E(r, E(s, m))* kepada pengirim (Alice)
3. Pengirim (Alice) mendekripsi pesan kedua dengan kunci *t*. Karena dari sifat komutatif dimana *D(t, E(r, E(s, m))) = E(r, m)* yang merupakan pesan dienkripsi dengan hanya

penerima *private key*. Lalu pengirim mengirimkan ini ke penerima.

D. Algoritma *Massey-Omura*

Massey-Omura Cryptosystem diusulkan oleh James Massey dan Jim K. Omura pada 1982 sebagai pengembangan atas *Three-Pass Protocol*.

Massey-Omura Cryptosystem didesain baik untuk pembelajaran dan pendidikan. Secara khusus, sistem *Massey-Omura* tradisional menggunakan *modulo exponential* dan bilangan prima. Dengan demikian, kriptografi dapat memanfaatkan matematika diskrit untuk menghitung kunci enkripsi dan dekripsi. Sistem yang dikembangkan dalam tugas akhir ini mengikuti algoritma *Massey-Omura* tradisional.

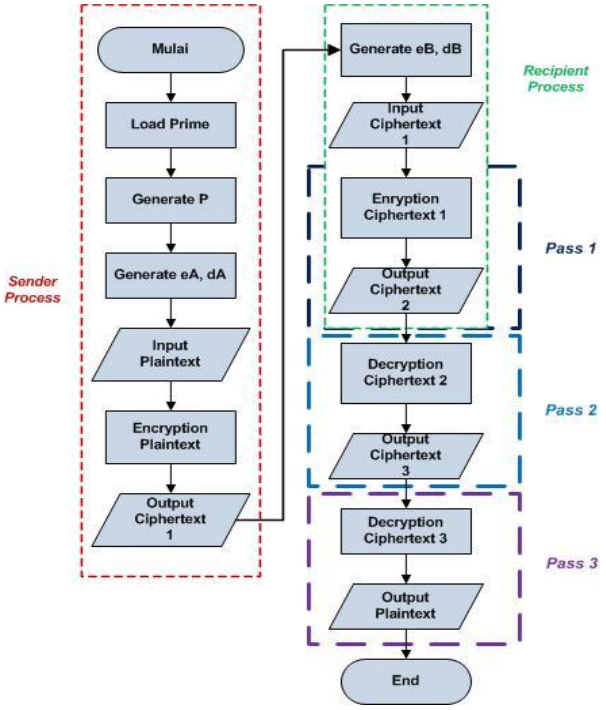
Berikut cara kerja dari Algoritma *Massey-Omura* :

- Semua pengguna telah menspakati kelompok batasan atas bidang tetap batasan *Fp* dengan *p* sebagai kekuatan utama
- Setiap pengguna secara rahasia memilih acak bilangan bulat *e* antara *0* dan *p – 1* seperti *GCD (e, p – 1) = 1*, dan menghitung $d = e^{-1} \bmod (p - 1)$ dengan menggunakan algoritma *euclidean*.
- Sekarang anggaplah bahwa Alice ingin mengirim pesan *M* yang aman untuk Bob, kemudian mereka ikuti prosedur berikut :

  1. Alice pertama mengirimkan M^{eA} kepada Bob,
  2. Pada saat menerima pesan, Bob mengirimkan M^{eAeB} kembali ke Alice (perhatikan bahwa saat ini, Bob tidak bisa membaca pesan Alice *M*)
  3. Alice mengirim $M^{eAeBdA} = M^{eB}$ kepada Bob, Bob kemudian menghitung $M^{dBeB} = M$, dan terbukalah pesan asli Alice *M*

E. Perancangan *Flowchart* Sistem

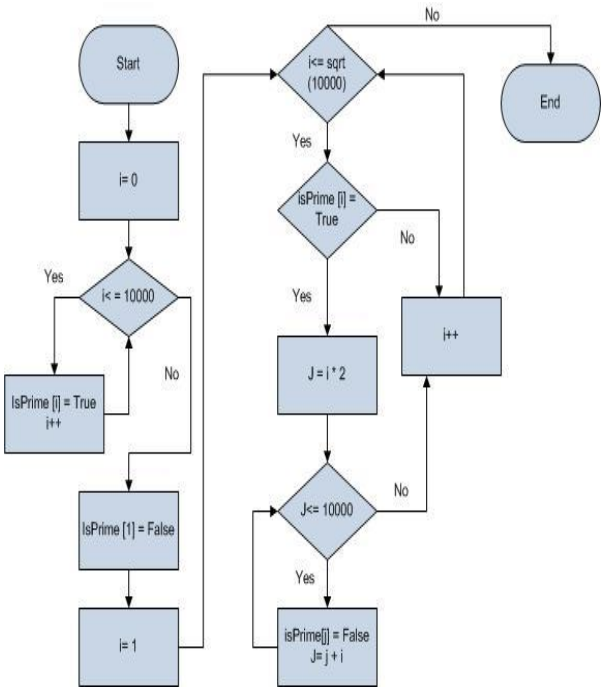
Perancangan *flowchart* merupakan rancangan alur proses yang ada dalam program simulasi. *Flowchart* menjelaskan tampilan *user interface*, proses pembangkitan kunci, proses enkripsi, dan proses dekripsi dalam program simulasi.



Gambar 2 *Flowchart User Interface*

Proses pertama kali yang dilakukan dalam sistem adalah proses pembangkitan kunci. Masalah utama dalam proses pembangkitan kunci adalah bagaimana menghasilkan kunci yang tidak dapat diprediksi. Ada berbagai metode yang dapat digunakan untuk menghasilkan sebuah bilangan prima yang besar. Salah satu metode yang dapat digunakan adalah dengan metode *The Sieve of Eratosthenes*.

Flowchart dari proses pembentukan kunci dapat dilihat pada gambar *flowchart testing primes* berikut ini:

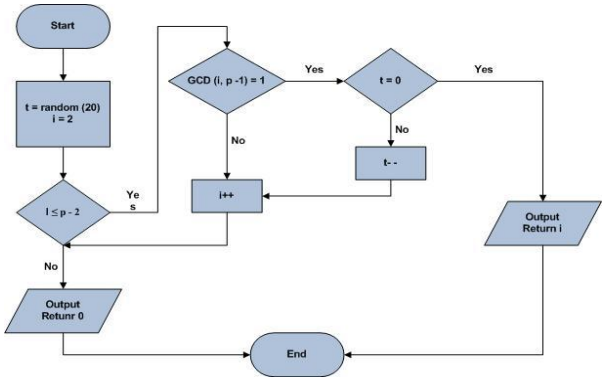


Gambar 3 *Flowchart testing primes*

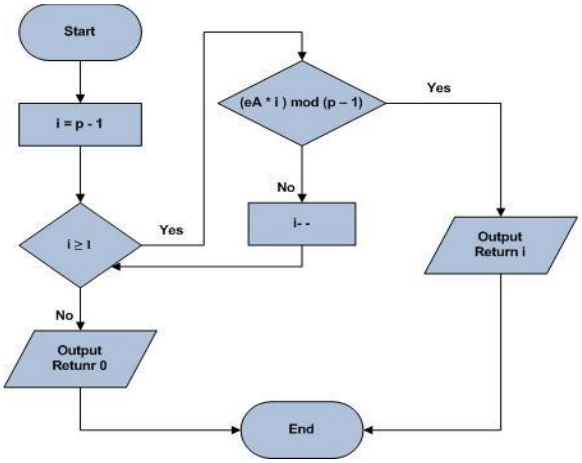
Proses yang harus dilakukan berikutnya adalah melakukan proses *generate key* enkripsi eA dan dekripsi dA (Pengirim).

Kemudian digunakan algoritma *Euclidean GCD* untuk mencari Faktor Persekutuan Terbesar (FPB) untuk eA.

Berikut merupakan *flowchart generate* enkripsi (eA) dan dekripsi (dA):



Gambar 4 *Flowchart generate* enkripsi pengirim (eA)



Gambar 5 *Flowchart generate* dekripsi pengirim (dA)

Setelah didapat *private key* eA dan dA, proses selanjutnya adalah melakukan proses enkripsi pesan *plaintext* menjadi *ciphertext 1* menggunakan teori *modulo exponential*.

Setelah proses enkripsi pesan dilakukan, pesan tersebut dikirim ke penerima (disimpan). Proses selanjutnya adalah melakukan *generate key* enkripsi (eB) dan dekripsi (dB) penerima. *Flowchart* dari proses ini sama dengan proses *generate key* enkripsi eA dan dekripsi dA pengirim.

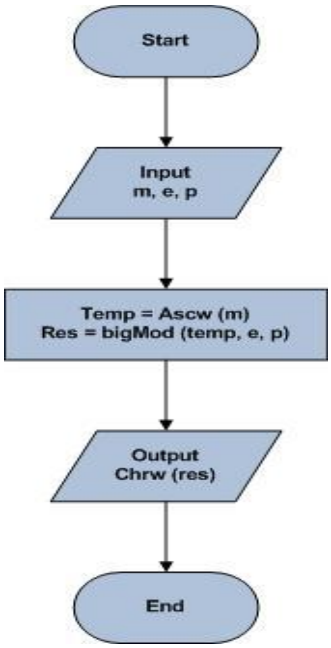
Setelah *private key* enkripsi dan dekripsi penerima didapat, proses selanjutnya adalah membuka pesan *ciphertext 1* dan kemudian melakukan proses enkripsi pesan *ciphertext 1* menjadi pesan *ciphertext 2* dengan menggunakan *private key* enkripsi (eB). *Flowchart* dari proses ini juga sama dengan proses enkripsi pesan pada pengirim.

Setelah pesan *ciphertext 2* didapat, pesan tersebut kemudian dikirim kembali ke pengirim pesan (disimpan). Pesan tersebut dibuka kembali untuk didekripsi oleh pengirim dengan menggunakan kunci dekripsi (dA) menjadi *ciphertext 3*. Setelah pesan tersebut didapat, pesan tersebut dikirim kembali ke penerima (disimpan).

Setelah pesan diterima oleh penerima, pesan tersebut didekripsi menggunakan kunci dekripsi (dB). Pesan *ciphertext 3* tersebut akan didekripsi menjadi *plaintext* kembali. Kemudian pesan tersebut disimpan.

Pada proses pengenkripsian dan pendekripsian *plaintext* → *ciphertext* dan *ciphertext* → *plaintext* dapat

dilihat pada gambar *flowchart generate cipher* berikut ini:



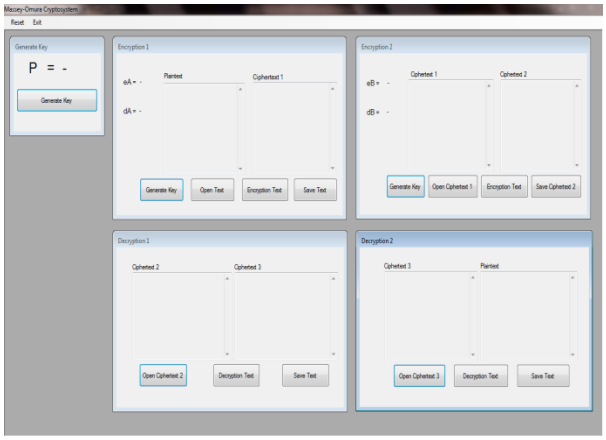
Gambar 6 *Flowchart generate cipher*

V. HASIL DAN DISKUSI

A. MDI Form

MDI Form merupakan form utama dari aplikasi. Pada MDI Form terdapat dua menu yakni *Reset* dan *Exit*. Dalam MDI form memiliki 5 form lagi : *Generate Key, Encryption 1, Encryption 2, Decryption 1, Decryption 2.*

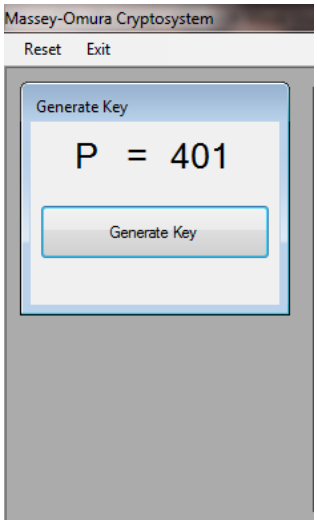
Menu Reset digunakan untuk mengulang dari awal seluruh isi dari beberapa form yang ada di dalam MDI Form. *Menu Exit* digunakan untuk keluar dari aplikasi.



Gambar 7 MDI Form

B. Form *Generate Key*

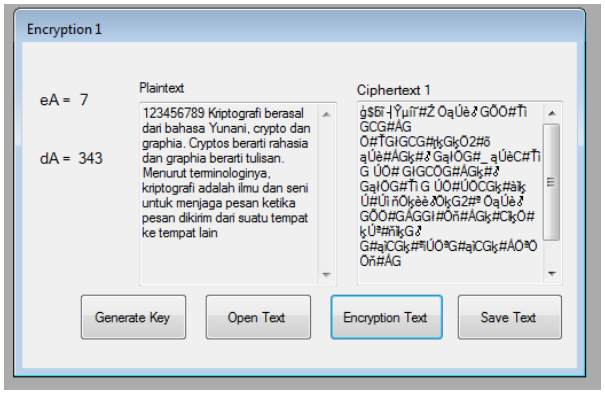
Langkah pertama yang dilakukan adalah melakukan *generate key p* pada form *generate key* dengan menekan *button generate key*.



Gambar 8 Form *generate key*

C. Form *Encryption 1*

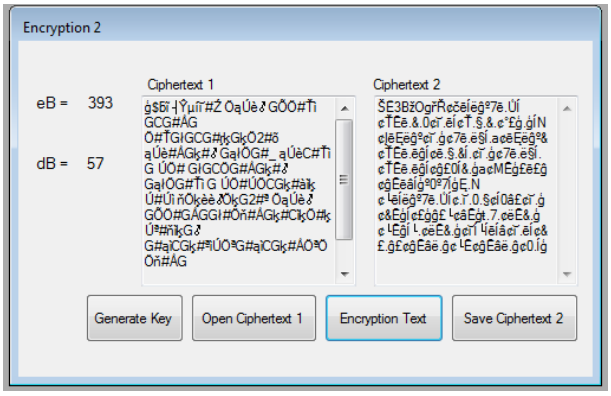
Pada form encryption 1 setelah nilai *p* didapat, cari nilai eA dan dA dengan menekan *button Generate Key Open Text* untuk membuka *file plaintext* (pesan asli). *Encryption Text* untuk melakukan proses enkripsi pesan. *Save Text* untuk menyimpan pesan.



Gambar 9 Form *Encryption 1*

D. Form *Encryption 2*

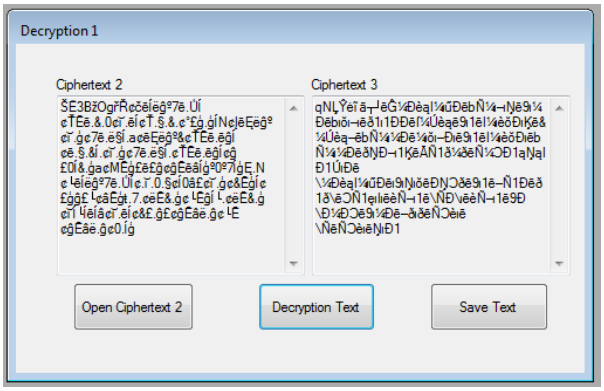
Pada form *encryption 2* tekan *button* generate key kembali untuk menampilkan nilai eB dan dB. *Open ciphertext 1* untuk membuka kembali pesan *ciphertext 1*. *Encryption text* untuk mengenkripsi *ciphertext 1* menjadi *ciphertext 2*. *Save text* untuk menyimpan pesan.



Gambar 10 Form *Encryption 2*

E. Form *Decryption 1*

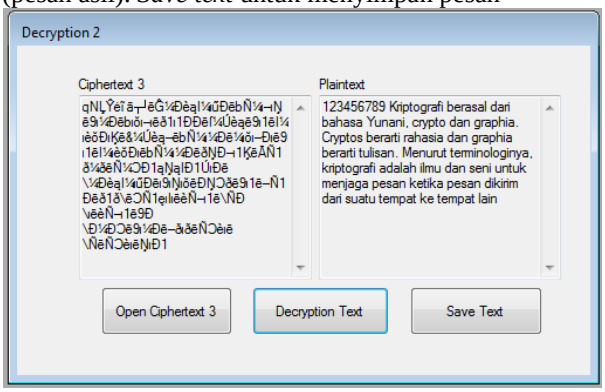
Pada form *decryption 1* tekan *button open ciphertext 2* untuk membuka kembali pesan *ciphertext 2*. *Decryption text* untuk mendekripsi *ciphertext 2* menjadi *ciphertext 3*. *Save text* untuk menyimpan pesan.



Gambar 11 Form *Decryption 1*

F. Form *Decryption 2*

Pada form *decryption 2* tekan *button open ciphertext 3* untuk membuka kembali pesan *ciphertext 3*. *Decryption text* untuk mendekripsi *ciphertext 3* menjadi *plaintext* (pesan asli). *Save text* untuk menyimpan pesan



Gambar 12 Form *Decryption 2*

VI. KESIMPULAN

Setelah melakukan studi literatur, perancangan, analisis, implementasi dan pengujian aplikasi untuk pengamanan *file* teks menggunakan algoritma *Massey-Omura* maka dapat disimpulkan bahwa telah diperoleh suatu sistem yang menggunakan algoritma *Massey-Omura* untuk pengamanan *file* dan sistem ini dapat mengenkripsi dan mendekripsi *file* (.txt). Adapun saran untuk pengembangan dan perbaikan dari sistem ini adalah penambahan banyak bilangan prima yang dihasilkan oleh perangkat lunak, dimana perangkat lunak ini hanya mampu membangkitkan bilangan prima sebanyak 66000 bilangan, dimana hal ini masih jauh dari standard kriptografi yang mengisyaratkan bilangan prima yang digunakan berjumlah 64 digit. Selain itu, enkripsi dan

dekripsi hanya bisa dilakukan pada *file* berekstensi .txt *(plaintext).* Diharapkan pada masa mendatang pengembangan perangkat lunak dapat mengenkripsi dan mendekripsi *file* selain *plaintext.* Untuk pengembangan selanjutnya juga diharapkan perangkat lunak dapat mengirimkan pesan yang telah terenkripsi dan terdekripsi melalui jaringan *internet* sehingga memungkinkan pemanfaatan perangkat lunak dalam berkomunikasi secara rahasia dan aman.

DAFTAR PUSTAKA

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

[2] Winton, R. 2007. *Enhancing the Massey-Omura Cryptosystem.*

[3] Ariyus, D. 2008. *Pengantar Ilmu Kriptografi (Teori, Analisis dan Implementasi).* Edisi ke-1. Yogyakarta: Andi.

[4] Karls, M. A. 2010. *Codes, Cipher, and Cryptography-An Honors Colloquium. Primus.*

[5] Alghazali, M. R. 2010. Sieve of Eratosthenes, Algoritma Bilangan Prima. Makalah. Bandung: Institut Teknologi Bandung.

[6] *Kanamouri et al.* 2009. Quantum Three-Pass Protocol : Key Distribution using Quantum.