PENERAPAN METODE ALGORITMA GOST PADA PERANCANGAN APLIKASI KRIPTOGRAFI

[1]Agung Moses Catur Satria, [2]Manorang Gultom, [3]Ricky Imanuel Ndaumanu
[1,3]Teknik Informatika, STMIK Widya Dharma, Pontianak
[2]Sistem Informasi, STMIK Widya Dharma, Pontianak
e-mail: [1]amcsatria@gmail.com , [2]manoranggtm@yahoo.com, [3]ricky_ndaumanu@ymail.com

Abstract

Security and confidentiality of data is a crucial aspect in the field of communication, especially communication using computer media. Science used to secure the data is cryptography. This research will be discussed in the GOST algorithm encrypt and decrypt a text file, implementation by using Visual Basic.Net 2013. The file will be analyzed the results of encryption and decryption based on file size and the time span during encryption and decryption. GOST Cryptography is the science that uses mathematical equations to encrypt and decrypt data. GOST encryption on the text using a randomization system character of plaintext into chiperteks. This allows the security of the data and information contained in the file not to be accessible to those who do not have the authority to access. To obtain the actual information contained in the file, decryption process must be done with the algorithm, calculation functions and the same key. The conclusion of the GOST algorithm has a way of working together, to perform encryption and decryption. This application is expected to be a means of security and confidentiality of the data that is appropriate to protect the data in terms of effective and efficiency in the process of encryption and decryption. Advice can be given is the application is expected to be a learning tool for the user as the knowledge of cryptographic algorithms GOST.

***Keywords**: Application, Cryptography, GOST Algorithm, Visual Basic.Net*

Abstrak

Keamanan dan kerahasiaan data merupakan aspek terpenting dalam bidang komunikasi, khususnya komunikasi yang menggunakan media komputer. Bidang ilmu pengetahuan yang digunakan untuk mengamankan data adalah kriptografi. Pada penelitian ini akan membahas algoritma GOST dalam mengenkripsi dan mendekripsi file teks, implementasinya dengan menggunakan bahasa pemrograman Visual Basic.Net 2013. File tersebut akan dianalisa hasil dari enkripsi dan dekripsi berdasarkan ukuran file dan rentang waktu selama proses enkripsi dan dekripsi. Kriptografi GOST merupakan ilmu pengetahuan yang menggunakan persamaan matematis untuk melakukan enkripsi dan dekripsi data. Enkripsi GOST pada teks menggunakan system pengacakan karakter dari plainteks menjadi chiperteks. Hal ini memungkinkan keamanaan data dan informasi yang termuat di dalam file untuk tidak dapat diakses oleh pihak yang tidak memiliki otoritas mengakses. Untuk memperoleh informasi sebenarnya yang termuat di dalam file, proses dekripsi harus dilakukan dengan algoritma, fungsi perhitungan dan kunci yang sama. Kesimpulan dari algoritma GOST tersebut memiliki cara kerja yang sama, dalam melakukan enkripsi maupun dekripsi. Aplikasi ini diharapkan dapat menjadi sarana keamanan dan kerahasiaan data yang tepat untuk melindungi data dari segi efektif dan keefisienan dalam proses enkripsi dan dekripsi. Saran yang dapat diberikan adalah aplikasi ini diharapkan dapat menjadi sarana pembelajaran bagi user sebagai pengetahuan tentang kriptografi yang menggunakan algoritma GOST.

Kata Kunci: Aplikasi, Kriptografi, Algoritma GOST, Visual Basic.Net.

2. PENDAHULUAN

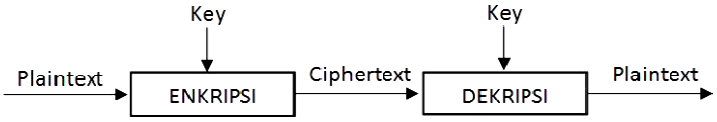
Keamanan dan kerahasiaan data merupakan aspek terpenting dalam bidang komunikasi, khususnya komunikasi yang menggunakan media komputer. Kriptografi merupakan ilmu untuk mengacak pesan demi keamanan data dan informasi di dalamnya. Persaingan dalam melakukan akses informasi memungkinan terjadinya kejahatan, seperti adanya pihak yang mengintip isi *file (sniffing)*, orang yang tidak memiliki otoritas menggandakan *file* atau informasi (*replay attack)* ataupun penyebaran informasi oleh orang yang bukan sebenarnya (*spoofing)*.

Penelitian ini membahas algoritma GOST dalam mengenkripsi dan mendekripsi *file* teks, pengumpulan data menggunakan penelitian dokumentasi dengan analisis sistem menggunakan *Unified Modeling Language* (UML), perancangan aplikasi menggunakan bahasa pemrograman Visual Basic.Net 2013.

Kriptografi algoritma GOST ini menggunakan persamaan matematis untuk melakukan enkripsi dan dekripsi data. Enkripsi algoritma GOST pada teks dalam aplikasi ini menggunakan sistem pengacakan karakter.

merubah plainteks menjadi cipherteks.. Hal ini memungkinkan keamanaan data dan informasi yang termuat di dalam *file* untuk tidak dapat dilihat oleh pihak yang tidak memiliki otoritas. Untuk mengembalikan data kebenrtuk semulaharus dilakukan proses dekripsi dengan algoritma, fungsi perhitungan, mode matematis dan kunci yang sama.

Kesimpulan dari algoritma GOST tersebut memiliki cara kerja yang sama, dalam melakukan enkripsi maupun dekripsi yaitu dengan menggunkaan proses dan perhitungan matematika, serta kunci yang sama. Aplikasi ini diharapkan dapat menjadi sarana keamanan dan kerahasiaan data yang tepat untuk melindungi data dari segi efektif dan keefisienan dalam proses enkripsi dan dekripsi. Saran yang dapat diberikan adalah aplikasi ini diharapkan dapat menjadi sarana pembelajaran bagi user sebagai pengetahuan tentang kriptografi yang menggunakan algoritma GOST. Secara umum, kriptografi digambarkan sbagai berikut:



Gambar 1. Skema Enkripsi dan Dekripsi Algoritma GOST

Penelitian ini melakukan perancangan aplikasi untuk melakukan enkripsi dan dekripsi *file* teks dan sebaliknya menggunakan algoritma *Government Standard* (GOST) atau standar pemerintah.

3. METODE PENELITIAN

3.1. Desain penelitian yang digunakan penulis adalah desain penelitian kausal (eksperimental) yaitu percobaan dan pengujian terhadap aplikasi yang dirancang.

3.1.1. Metode Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penyusunan penelitian ini: penelitian dokumentasi, yaitu penelitian dilakukan dalam bentuk menelusuri penggunaan literatur yang berkaitan dengan objek penelitian tersebut dari berbagai sumber, termasuk buku-buku pemrograman dasar, buku-buku kriptografi yang berhubungan dengan algoritma kriptografi GOST dan buku-buku analisis *Unified Modeling Language* (UML) dan dokumentasi serta pengujian dan pengecekan kesalahan (*error*) terhadap aplikasi yang telah dirancang.

3.1.2. Teknik Analisis Sistem

Teknik analisis sistem yang digunakan oleh penulis dalam penelitian menggunakan teknik analisis *Unified Modeling Language* (UML) untuk menvisualisasikan dan mendokumentasikan suatu sistem informasi agar lebih mendetail dan terperinci serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

3.1.3. Teknik Perancangan Aplikasi

Teknik perancangan aplikasi yang digunakan penulis dalam penelitian ini menggunakan bahasa pemrograman Microsoft Visual Studio Visual Basic.Net 2013.

3.2. Landasan Teori

3.2.1. Sistem

Sistem adalah sekumpulan komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan bersama. [1] Sistem merupakan kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin di capai.[2]

3.2.2. Perancangan Sistem.

Perancangan Sistem dapat didefinisikan sebagai rancangan aktivitas yang terdiri dari rancangan logika dan rancangan fisik, keduanya menghasilkan spesifikasi sistem yang memenuhi persyaratan sistem yang dikembangkan dalam tahap analisa sistem. [3]

3.2.3. Perangkat Lunak

Perangkat lunak (*software*) berwujud program untuk menjalankan perangkat keras komputer yang dapat dikelompokkan menjadi dua jenis, yaitu perangkat lunak sistem operasi (*operating system software*) dan perangkat lunak aplikasi (*application software*).[4]

3.2.4. Aplikasi

Aplikasi adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan *(User Manual)*.[5] Aplikasi merupakan program komputer sebagai sarana interaksi antara pengguna dan perangkat keras.[6]

3.2.5. Kriptografi

Kriptografi adalah ilmu untuk menjaga isi data atau pesan agar tetap aman. Aman disini berarti tidak bisa atau sulit diakses oleh orang lain yang tidak berhak.[7] Kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. [8]

3.2.6. Algoritma GOST

Algoritma GOST merupakan suatu algoritma *block cipher* yang dikembangankan oleh seorang berkebangsaan Uni Soviet untuk menyembunyikan data atau informasi yang bersifat rahasia pada saat komunikasi. [9] GOST merupakan blok kode 64 bit dengan panjang kunci 256 bit. Algoritma ini mengiterasi algoritma enkripsi sederhana sebanyak 32 putaran. Untuk mengenkripsi, pertama-tama teks asli 64 bit dipecah menjadi 32 bit bagian kiri, L dan 32 bit bagian kanan, R. subkunci untuk putaran *i* adalah K*i*. Pada satu putaran ke-*i*, operasinya adalah sebagai berikut:

$$Li = Ri\text{-}1$$
$$Ri = Li\text{-}1 \text{ xor } f(Ri\text{-}1, Ki)$$

Sedangkan pada fungsi *f*, mula-mula bagian kanan data ditambah dengan upa-kunci ke-*i* modulus 2^{32} . Hasilnya dipecah menjadi 8 bagian 4 bit dan setiap bagian menjadi masukan kotak-S yang berbeda. Didalam GOST terdapat 8 buah kotak-S kedua, dan seterusnya. Ouput dari 8 kotak-S kemudian dikombinasikan menjadi bilangan 32 bit kemudian bilangan ini di rotasi 11 bit ke kiri. Akhirnya hasil operasi ini di XOR dengan data bagian kiri yang kemudian menjadi bagian kanan dan bagian kanan menjadi bagian kiri(*swap*).[10]

3.2.7. Algoritma Simetri

Algoritma simetri sering disebut algoritma klasik karena memakai kunci yang sama untuk melakukan enkripsi dan dekripsi. [11] Algoritma Simetri merupakan teknik simetri berarti kunci ataupun kode untuk melakukan enkripsi sama dengan kunci atau kode untuk melakukan dekripsi.[12]

3.2.8. Kode Blok (Block Cipher)

Block cipher merupakan sebuah fungsi yang memetakan n-bit blok *plaintext* ke n-bit blok *ciphertext* ke dalam blok-blok yang cukup besar (≥64). [13] Kode blok *(block cipher)* merupakan algoritma yang masukan dan keluarannya merupakan satu blok dan setiap blok terdiri dari banyak blok. [14]

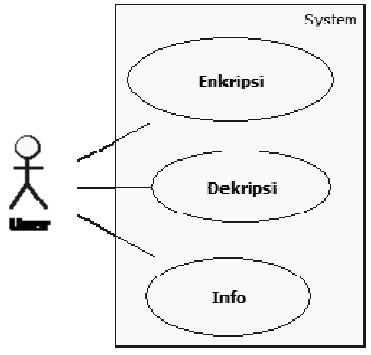
3.2.9. Microsoft Visual Basic.Net

Visual Basic.Net adalah salah satu program berorientasi objek, selain itu ada pula program java dan C++ yang juga berbasis objek.[15]

4. HASIL DAN PEMBAHASAN

Gambaran umum perancangan Aplikasi Kriptografi Algoritma GOST menggunakan *Unified Modeling Language* (UML)

4.1. Gambar umum Diagram Use Case Aplikasi Kriptografi Algoritma GOST



Gambar 2. Diagram Use Case Menu Utama

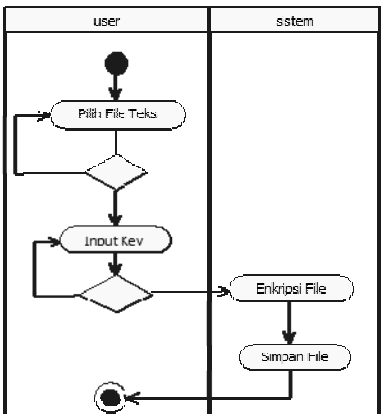
Diagram *use case* digunakan untuk menangkap perilaku dari aplikasi yang dibuat. Dalam diagram *use case*, *user* berperan penting dalam hubungannya dengan sistem. *User* berupa manusia. *Use case diagram* menggambarkan pemodelan yang digunakan untuk menggambarkan model dari sebuah aplikasi dimana *user* sebagai aktor yang berinteraksi dengan aplikasi kriptografi ini. *Use case diagram* mengidentifikasikan

fungsionalitas yang disediakan oleh sistem (*use case*) dan pengguna *(user*) yang saling berinteraksi dengan sistem.

4.2. Gambar Diagram Aktivitas

Diagram ini memperlihatkan aliran kerja dari mulainya suatu aktivitas hingga aktivitas berhenti di dalam suatu sistem. Berikut perancangan diagram aktivitas yang digunakan untuk perancangan aplikasi:

4.2.1. Gambar Diagram Aktivitas Enkripsi

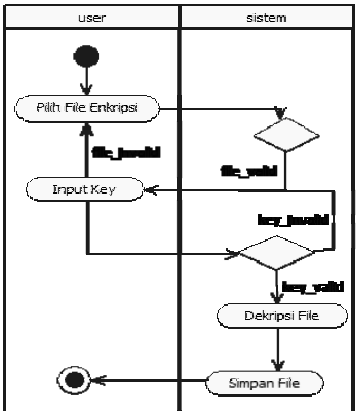


Gambar 3. Diagram Aktivitas Enkripsi

Diagram ini memperlihatkan aliran kerja dari mulainya suatu aktivitas hingga aktivitas berhenti di dalam suatu sistem. Berikut perancangan diagram aktivitas yang digunakan untuk perancangan aplikasi: Pengguna harus memilih *file* teks target yang akan dienkripsi, system akan memeriksa apakah *file* yang dipilih benar *file* jenis teks. Apabila validasi benar, maka sistem akan melanjutkan ke proses selanjutnya yaitu pengguna diharuskan dan memilih *directory* tempat menyimpam *file* hasil dari proses enkripsi di simpan dan meng-*input*-kan kunci *(key)*. Kunci akan divalidasi jenis dan jumlah karakter yang di-*input*-kan, bila sesuai dengan ketentuan sistem, maka pengguna menekan tombol enkripsi untuk melakukan eksekusi proses yang akan dilakukan oleh sistem.

4.2.2. Gambar Diagram Aktivitas Dekripsi

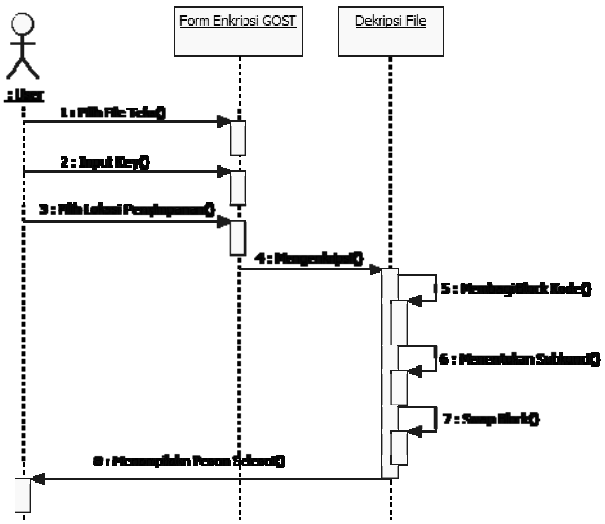
Activity diagram merupakan pemodelan untuk menggambarkan aktivitas yang terjadi pada aplikasi kriptografi. Dengan *activity diagram* dapat dilihat jalannya proses yang akan terjadi saat aplikasi dijalankan.



Gambar 4. Diagram Aktivitas Dekripsi

Pengguna harus memilih *file* teks target yang akan didekripsi, sistem akan memeriksa apakah *file* yang dipilih benar *file* jenis yang telah dienkripsi. Apabila validasi benar *file* yang dipilih adalah *file* terenkripsi, maka sistem akan melanjutkan ke proses selanjutnya yaitu pengguna diharuskan memilih *directory* tempat menyimpam *file* hasil dari proses enkripsi di simpan, menentukan jenis ekstensi *file* dan meng-*input*-kan kunci *(key)*. Kunci akan divalidasi jenis dan jumlah karakter yang di-*input*-kan, bila sesuai dengan ketentuan sistem, maka pengguna menekan tombol enkripsi untuk melakukan eksekusi proses yang akan dilakukan oleh sistem.

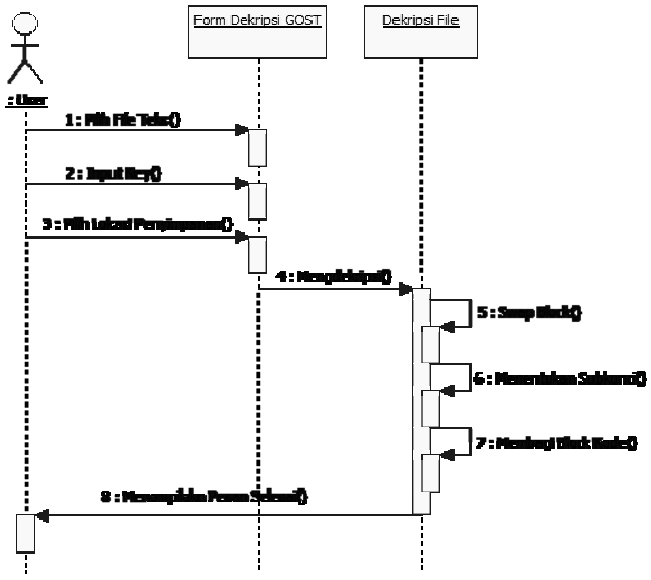
4.3. Gambar Diagram Sekuensial
4.3.1. Gambar Diagram Sekuensial Enkripsi



Gambar 5. Diagram Sekuensial Enkripsi

Pengguna masuk *form* utama, memilih *form* enkripsi, masuk ke dalam *form* enkripsi, kemudian pengguna memilih *file* teks target. Setelah memilih, pengguna memilih *directory* tempat menyimpam hasil enkripsi, kemudian meng-*input*-kan kunci dan dilanjutkan proses enkripsi algoritma GOST di dalam aplikasi yaitu membagi blok kode, menentukan subkunci, melakukan perhitungan, melakukan pengacakan karakter, melakukan perhitungan dengan metode perhitungan, melakukan konversi nilai dan terakhir melakukan *swap* blok. Setelah selesai, aplikasi akan menampilkan pesan bahwa proses enkripsi telah selesai.

4.3.2. Gambar Diagram Sekuensial Dekripsi



Gambar 5. Diagram Sekuensial Dekripsi

Pengguna masuk *form* utama, memilih *button* dekripsi untuk masuk ke *form* dekripsi, memilih *file* target yang telah terenkripsi untuk didekripsi. Pengguna meng-*input*-kan *key* yang sama saat melakukan enkripsi, memilih *directory* penyimpan *file*, proses dekripsi didalam algoritma GOST berjalan yaitu kebalikan dari proses enkripsi, yaitu melakukan *swap* blok, menentukan subkunci dan mengembalikan *block code*. Setelah selesai, aplikasi akan menampilakan pesan selesai kepada *user* bahwa proses dekripsi telah selesai.

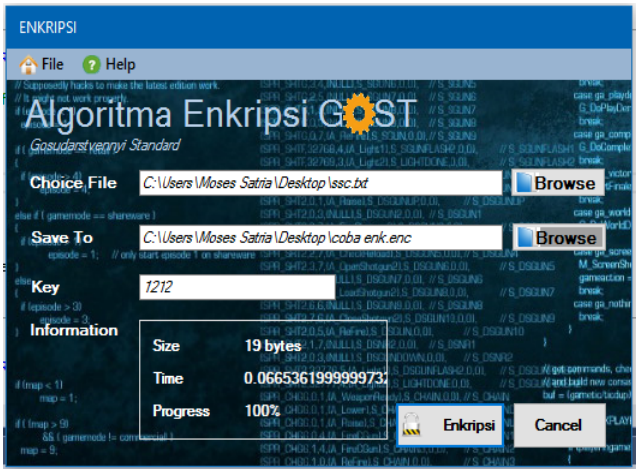
4.4. Tampilan Aplikasi
4.4.1. Tampilan Form Utama

Form utama menampilkan halaman utama aplikasi. Pengguna dapat memilih untuk melakukan enkripsi dengan memilih dan menekan tombol ekripsi atau memilih melakukan dekripsi *file* terenkripsi dengan

melakukan dan menekan tombol dekripsi dekripsi. Bila tombol enkripsi ditekan maka akan menampilkan *form* dekripsi dan akan menampilkan pilihan untuk diisi agar dapat melakukan proses dekripsi dengan kelengkapan data yang akan di olah berupa *file* teks, atau dapat melalui *button* atau memilih melaui menu *file.* Pengguna dapat mengakses bantuan dan informasi mengenai program dengan memilih menu info serta keluar dari aplikasi dengan *button* atau menu *file. Form* utama hanya memuat pilihan utama memberikan dua pilihan untuk melakukan proses, selain itu pilihan juga dapat dipilih dengan menu *file.*

Gambar 7. Tampilan Form Utama

4.4.2. Tampilan Form Enkripsi



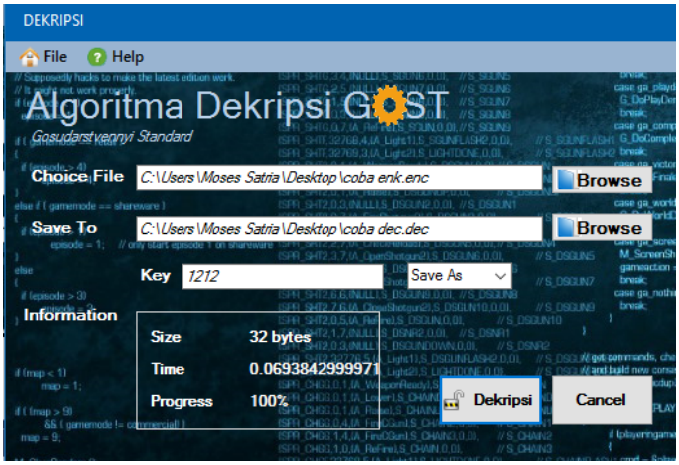
Gambar 7. Tampilan Form Enkripsi

Form enkripsi menampilkan beberapa komponen yang mengharuskan pengguna mengisi dan melengkapi setiap textbox, menekan tombol-tombol dan memasukkan kunci ke dalam textbox untuk dapat melakukan proses enkripsi algoritma GOST, pengguna menentukan file sumber yang dipilih dengan menekan tombol browse yang terletak di samping textbox choice file. Textbox nantinya akan terisi dengan url tempat drive dan file dipilih dan ditentukan untuk melengkapi proses di dalam algoritma enkripsi, menentukan directory tempat penyimpanan hasil enkripsi dengan menekan tombol browse di samping textbox save to. Textbox save to akan menampilkan url drive tempat penentuan folder dan letak file berada, kemudian meng-input-kan kunci pada textbox key. Textbox key tidak menampilkan karakter string melainkan hanya menampilkan karakter password sebagai penyembunyian dari pihak yang tidak diinginkan. Setelah semua dilengkapi, pengguna menekan tombol Enkripsi dan proses enkripsi berjalan di dalam aplikasi. Aplikasi akan menampilkan pemberitahuan proses telah selesai setelah aplikasi melakukan eksekusi algoritma GOST melakukan enkripsi. Informasi akan di tampilkan di groupbox information dengan menampilkan size, time dan progress kerja aplikasi untuk dapat dilihat pengguna. Size menampilkan ukuran file setelah proses enkripsi, time menampilkan waktu aplikasi meakukan enkripsi, progress menampilkan persentase aplikasi melakukan proses enkripsi.

4.4.3. Tampilan Form Dekripsi

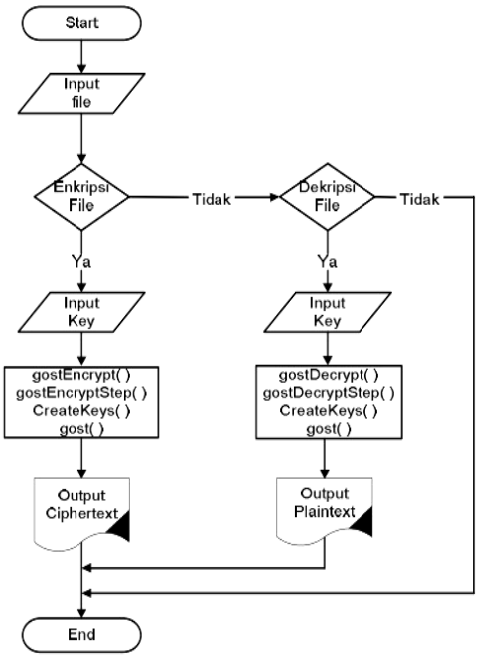
Form dekripsi menampilkan beberapa komponen yang mengharuskan pengguna mengisi dan melengkapi setiap *textbox,* menekan tombol-tombol dan memasukkan kunci *(key)* ke dalam *textbox* untuk dapat melakukan

proses dekripsi algoritma GOST, pengguna menentukan *file* sumber yang dipilih dengan menekan tombol *browse* yang terletak di samping *textbox choice file*. *File* yang dipilih merupakan *file* yang telah terenkripsi. *Textbox* nantinya akan terisi dan menampilkan url tempat *drive* dan *file* yang telah dipilih dan ditentukan untuk melengkapi proses di dalam algoritma enkripsi, menentukan *directory* tempat penyimpanan hasil enkripsi dengan menekan tombol *browse* di samping *textbox save to*. *Textbox save to* akan menampilkan url *drive* tempat penentuan *folder* dan letak *file* berada, kemudian meng-*input*-kan kunci pada *textbox key*. *Textbox key* tidak menampilkan karakter string melainkan hanya menampilkan karakter password sebagai penyembunyian dari pihak yang tidak diinginkan. Kunci yang dimasukkan harus sama persisi dengan kunci saat melakukan enkripsi. Setelah semua dilengkapi, pengguna menekan tombol dekripsi dan proses dekripsi berjalan di dalam aplikasi. Aplikasi akan menampilkan pemberitahuan proses telah selesai setelah aplikasi melakukan eksekusi algoritma GOST melakukan dekripsi. Informasi akan di tampilkan di *groupbox information* dengan menampilkan *size*, *time* dan *progress* kerja aplikasi untuk dapat dilihat pengguna. *Size* menampilkan ukuran *file* setelah proses dekripsi, *time* menampilkan waktu aplikasi meakukan dekripsi, *progress* menampilkan persentase aplikasi melakukan proses dekripsi.



Gambar 8. Tampilan Form Dekripsi

Proses enkripsi dan dekripsi algoritma GOST dapat dilihat pada gambar berikut:



Gambar 9. Enkripsi dan Dekripsi Algoritma GOST

4.5. Langkah-langkah Proses Algoritma GOST

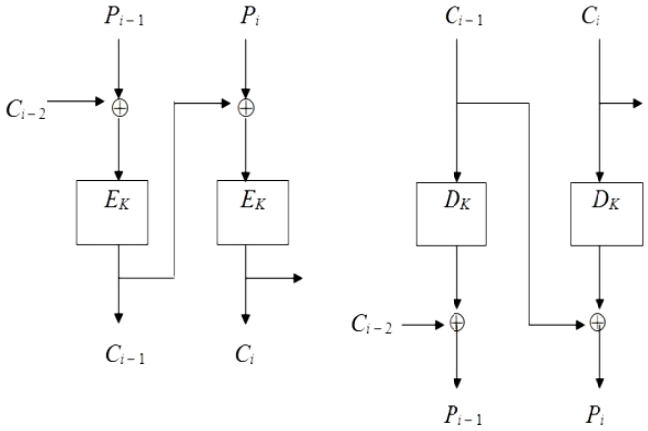
4.5.1. Proses Enkripsi

a. Langkah pertama sebelum memulai putaran, algoritma terlebih dahulu mendapatkan nilai biner dari tiap katakter didalam *file* dengan cara aplikasi mengkonversi plainteks ke nilai biner. Hasil konversi menghasilkan empat blok bit biner. Satu blok bit biner memiliki 8 nilai. Jumlah blok kode dalam satu blok kode terdapat 64 bit biner.
b. Setelah didapatkan nilai biner 8 bit, kemudian melakukan proses putaran. Dimulai dari proses satu putaran, putaran ke-0.
c. Langkah kedua yaitu nilai biner 64 bit plainteks dipecah menjadi 2 bagian. 32 bit bagian kiri (L) dan 32 bit bagian kanan (R).
d. Langkah ketiga yaitu fungsi f . Fungsi f merupakan fungsi yang memproses 32 bit bagian kanan (R) ditambahkan dengan kunci (K) modulus 32 dengan rumus : R(0) + K(0) mod 2^32.
e. Langkah keempat, biner 32 bit dipecah menjadi 8 kelompok masing-masing 4 bit. Dan dimasukkan ke dalam tabel S-BOX. Di dalam GOST terdapat 8 buah S-BOX. 4 bit pertama menjadi S-BOX pertama, 4 bit kedua menjadi S-BOX kedua, dan seterusnya.

Tabel *S-Box*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
***S-Box* 0**	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
***S-Box* 1**	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
***S-Box* 2**	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
***S-Box* 3**	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
***S-Box* 4**	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
***S-Box* 5**	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
***S-Box* 6**	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
***S-Box* 7**	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Gambar 10 S-BOX

f. Langkah kelima, hasil keluaran (*output*) biner yang didapat dari substitusi S-BOX digabungkan kembali dengan bilangan 32 bit dan lakukan pergeseran (*Rotate Left Shift / Swap*) sebanyak 11 bit ke kiri.
g. Langkah keenam, proses untuk mendapatkan R(1) = R(0) XOR L(0). Proses keenam untuk mendapatkan nilai bagian kanan R(1) yang nantinya akan digunakan untuk proses putaran selanjutnya. Di dalam proses XOR ada metode perhitungan yang telah ditentukan. Dalam algoritma penelitian ini, metode perhitungan yang digunakan adalah mode *Cipher Block Chaining* (CBC) yaitu mekanisme umpan balik *(feedback)* ke dalam proses enkripsi blok bit, dimana hasil enkripsi diumpanbalikan ke dalam blok *current* yang dioperasikan pada bit kelompok atau blok bit. Skema mode CBC dapat dilihat pada gambar berikut:



Gambar 11. Mode Cipher Block Chaining (CBC)

h. Langkah ketujuh mendapatkan L(1). Nilai L(1) didapat dengan mengambil nilai R(0) sebelum proses. Yaitu saat 64 bit dipecah menjadi dua bagian 32 bit dan mengambil nilai 32 bit R(0).
i. Langkah kedelapan, pada putaran ke-30 (putaran akhir) langkah 7 dan 8 berbeda. Yaitu pada putaran ke-31 langkah ketujuh mengembalikan nilai *string* untuk nilai R(31) dan L(31)
j. Langkah kesembilan yaitu menggabungkan dua nilai biner menjadi satu dengan rumus R(31) dan L(31).
k. Langkah kesebelas yaitu hasil biner 64 bit dari R(31) dan L(31) dipecah kembali menjadi 8 kelompok masing-masing 8 bit. Setelah itu dikonversi ke nilai *American Standard Code information for Interchange* (ASSCI).

4.5.2. Proses Dekripsi

Proses dekripsi merupakan kebalikan dari proses enkripsi. Proses dekripsi dimulai dari proses konversi nilai ASSCI, pemecahan biner 64 bit R dan 64 bit L menjadi delapan kelompok ke dalam S-BOX, proses perhitungan CBC, proses XOR, pembagian blok bit biner, pengurutan dan penjadwalan kunci, perputaran (*rotate*), pertukaran bit (*swap*), penggabungan blok bit kembali, penggabungan blok kode, hingga ke langkah pertama proses enkripsi yaitu mengkonversi biner kembali ke string.

4. KESIMPULAN

Aplikasi yang dihasilkan dengan menerapkan algoritma kriptografi GOST ini mampu melakukan proses eksekusi enkripsi dan dekripsi *file* dengan baik, cepat dan hasil dari enkripsi menghasilkan karakter acak yang sulit untuk dibaca karena benar-benar karakter acak sehingga sulit untuk dapat memperoleh, membaca dan mendapatkan isi dari *file* tersebut serta isi informasi yang termuat didalamnya harus melakukan proses dekripsi dengan aplikasi, proses dekripsi menghasilkan teks yang sama dengan teks asli sebelum proses enkripsi. Proses dekripsi harus menggunakan algoritma dan kunci yang sama dengan saat melakukan proses enkripsi. Aplikasi ini sangat mempuni untuk menangani pengamanan *file* teks, serta memiliki kerumitan dan kunci yang simetris hal ini untuk menjadi sarana keamanan *file* teks untuk tetap menjaga kerahasiaan informasi dan otentikasi data yang terdapat di dalam *file*.

Berdasarkan penjelasan pada bab satu sampai dengan bab empat sebelumnya, dapat diambil kesimpulan mengenai Aplikasi Kriptografi Algoritma GOST, yaitu :

a. Aplikasi Kriptografi Algoritma GOST dapat membantu pengguna menangani masalah keamanan informasi di dalam *file* untuk tetap menjaga kerahasiaan dari pihak yang tidak memiliki otoritas mengakses *file*.
b. Dengan adanya aplikasi Algoritma Kriptografi GOST ini, pengguna hanya membutuhkan sebuah laptop dan waktu singkat untuk menyembunyikan informasi yang ingin dirahasiakan. Karena proses enkripsi dilakkan dengan cepat.
c. Aplikasi yang dibangun memiliki *interface* yang *user friendly*. Hal ini memudahkan pengguna dalam pengoperasian aplikasi.
d. Aplikasi yang dibangun berbasis *stand alone*, sehingga dapat digunakan dimana saja hanya dengan melakukan proses instalasi dalam waktu singkat, tanpa harus takut informasi dalam *file* dapat diketahui pihak yang tidak memiliki otoritas.
e. Aplikasi tidak memiliki proteksi dan database tersisip di dalam *file* terenkripsi untuk menyimpan kunci bila terjadi kesalahan pengguna seperti lupa kunci.
f. Tidak ada pembatasan usia untuk penggunaannya.

5. SARAN

Dari perancangan aplikasi kriptografi algoritma GOST ini, , diharapkan dapat menjadi dasar penelitian lebih lanjut, mengingat banyaknya keterbatasan yang dihadapi oleh penulis, maka diusulkan beberapa saran dalam pengembangan, yaitu:

a. Aplikasi dapat dikembangkan dengan menambahkan algoritma dan metode kriptografi lain setelah proses enkripsi ataupun metode keamanan data lainnya.
b. Aplikasi dapat dikembangkan untuk melakukan dan menangani kriptografi enkripsi dan dekripsi untuk jenis data lainnya, seperti *audio, video dan image*.
c. Aplikasi dapat ditambahkan detail-detail lainnya guna meningkatkan kelengkapan aplikasi.
d. Aplikasi dapat ditambahkan proteksi penanggulangan lupa kunci yang memungkinkan pengguna dapat mendekripsi file yang telah terenkrip dan lupa kunci.
e. Untuk meningkatkan interaksi antara pengguna dengan aplikasi diharapkan agar dibuat desain *interface* yang lebih menarik dan *user friendly* untuk memudahkan pengguna dalam menggunakan aplikasi.

UCAPAN TERIMA KASIH

Dalam penelitian ini, penulis telah banyak mendapat bantuan dari berbagai pihak yang turut membantu, baik berupa bimbingan, petunjuk, saran, dorongan moril, dan doa. Maka pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya terutama kepada kedua orang tua yang telah memberikan dukungan finansial selama saya menyusun penelitian ini, dan kepada seluruh civitas akedemika sekolah tinggi manajemen informatika dan komputer widya dharma pontianak.

DAFTAR PUSTAKA

[1] Supriyanto, Wahyu, dan Ahmad Muhsin. (2008). *Teknologi Informasi Perpustakaan*. Kanisius. Yogyakarta.
[2] Rosa, A. S., dan Shalahuddin. (2015). Rekayasa *Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika. Bandung.
[4] Mulyanto, Aunur. R. (2008). *Rekayasa Perangkat Lunak*. Jilid 1. Departeman Pendidikan Nasional. Jakarta.
[5] Arryawan, Eko. (2010). *Anti Forensik.* PT Elex Media Komputindo. Jakarta.
[6] Ariyus, Dony. (2008). *Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi*. Andi offset. Yogyakarta.
[7] Wardana. (2008). *Membuat Aplikasi Berbasis Pendekatan Sistem Dengan Visual Basic Net*. PT Elex Media Komputindo. Jakarta.
[8] Sanusi, Muzammil. (2010). *The Genius Hacking Sang Pembobol Data*. PT Elex Media Komputindo. Jakarta.
[9] Ariyus, Dony. (2008). *Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi*. Andi offset. Yogyakarta.
[10] Wardana. (2008). *Membuat Aplikasi Berbasis Pendekatan Sistem Dengan Visual Basic Net*. PT Elex Media Komputindo. Jakarta.