Perancangan dan Implementasi Enkripsi dan Dekripsi File dengan Algoritma RC4 – One Time Pad pada Jaringan LAN

Pandapotan Sihombing [1], Wasit Ginting [2]

[1,2] *Universitas Katolik Santo Thomas Medan, Jl. Setiabudi No. 479 F. Tanjungsari, Medan, Indonesia*

ARTICLE INFORMATION

Received: Februari, 20, 2020
Revised: Maret, 2, 2020
Available online: April,9 2020

KEYWORDS

Five words maximum, comma separated

ENKRIPSI, DEKRIPSI, RC4, ONE TIME PAD, JARINGAN LAN DAN KRIPTOGRAFI

Phone: +62 813-7691-5539

E-mail: mmcbyes@gmail.com

A B S T R A K

Sekarang ini keamanan yang efektif dari suatu sistem sangat diperlukan untuk kegiatan bisnis sehari-hari. Sistem yang aman bisa memberikan tingkat kepercayaan yang tinggi kepada pengguna sehingga bisa memberi nilai tambah dan daya guna bagi sistem itu sendiri. Pengguna akan merasa nyaman dan aman ketika berhubungan dengan sistem yang bisa mengamankan data pengguna dari penyerang. Penelitian ini membahas tentang bagaimana mengenkripsi dan mendekripsi file dengan menggunakan metode RC4 dan One Time Pad, dimana file yang di enkripsi dapat dikirim dari client yang satu ke client yang lain atau ke server. Dengan menggunakan metode/aplikasi ini maka file yang dikirim akan terjamin keamanannya.

PENDAHULUAN

Kemajuan teknologi di bidang komputer memungkinkan ribuan orang dan komputer di seluruh dunia terhubung dalam satu dunia maya yang dikenal sebagai *cyberspace* atau Internet. Begitu juga ratusan organisasi seperti perusahaan, lembaga negara, lembaga keuangan, militer dan sebagainya. Tetapi kemajuan teknologi ini selalu diikuti dengan sisi buruk dari teknologi itu sendiri. Salah satunya adalah rawannya keamanan data sehingga menimbulkan tantangan dan tuntutan akan tersedianya suatu sistem pengamanan data yang sama canggihnya dengan kemajuan teknologi komputer itu sendiri. Ini adalah latar belakang berkembangnya sistem keamanan data untuk melindungi data yang ditransmisikan melalui suatu jaringan komunikasi.

File adalah kumpulan catatan atau arsip data baik berupa teks, angka, gambar, video, program dan lain-lain yang diberi nama tertentu secara digital. File bisa diterjemahkan sebagai arsip maupun data yang tersimpan di dalam komputer. Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berujut suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa digunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep[1].

Keamanan yang efektif dari suatu sistem sangat diperlukan untuk kegiatan bisnis sehari-hari. Sistem yang aman bisa memberikan tingkat kepercayaan yang tinggi kepada pengguna sehingga bisa memberi nilai tambah dan daya guna bagi sistem itu sendiri. Pengguna akan merasa nyaman dan aman ketika berhubungan dengan sistem yang bisa mengamankan data pengguna dari penyerang[2], [3].

Dalam teknologi informasi telah dan sedang dikembangkan cara-cara untuk menangkal berbagai serangan, seperti penyadap dan pengubahan data yang sedang dikirimkan. Salah satu cara yang ditempuh untuk mengatasi masalah ini adalah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak yang tidak berhak mengakses. Transformasi ini memberikan solusi pada dua macam masalah keamanan data, yaitu masalah privasi (*privacy*) dan keotentikan (*authentication*) [4], [5].

Jaringan komputer adalah sekelompok komputer otonom yang dihubungkan satu dengan lainnya dengan menggunakan protokol komunikasi melalui media transmisi atau media komunikasi sehinga dapat saling berbagi data-informasi, program-program, penggunaan bersama perangkat keras seperti printer, harddisk, dan CPU. Jaringan komputer adalah himpunan "interkoneksi" antara 2 komputer autonomous atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Sebuah komputer dapat membuat komputer lainnya restart, shutdown, atau melakukan kontrol lainnya, maka komputer-komputer tersebuat bukan *autonomous* (tidak melakukan kontrol terhadap komputer lain dengan akses penuh)[6], [7].

Jaringan komputer merupakan sekumpulan komputer yang saling berhubungan dan dapat saling berbagi informasi pada suatu jaringan komputer. Sebuah jaringan komputer paling sedikit terdiri dari dua komputer yang saling terhubung dengan sebuah media sehingga komputer-komputer tersebut dapat saling berbagi *resource* dan saling berkomunikasi [8].

RC4 merupakan enkripsi stream simetrik proprietary yang dibuat oleh RSA Data Security Inc (RSADSI). Penyebarannya diawali dari sebuah source code yang diyakini sebagai RC4 dan dipublikasikan secara 'anonymously' pada tahun 1994. Algoritma yang dipublikasikan ini sangat identik dengan implementasi RC4 pada produk resmi. RC4 digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman[9]. Sampai saat ini diketahui tidak ada yang dapat memecahkan/membongkarnya, hanya saja versi ekspor 40 bitnya dapat dibongkar dengan cara "brute force" (mencoba semua kunci yang mungkin)[10]. RC4 tidak dipatenkan oleh RSADSI, hanya saja tidak diperdagangkan secara bebas (*trade secret*).

Metode enkripsi one-time pad merupakan salah satu golongan metode enkripsi kunci simetris, kategori stream cipher. Metode ini sering disebut Vernam Cipher, yang merupakan metode enkripsi kunci simetris yang tidak terpecahkan (*unbreakable by exhaustive search*). Metode enkripsi one-time pad merupakan metode yang sempurna (*perfect methods*) yang paling sederhana [11], [12].

Algoritma yang dapat mengamankan data yang dibahas adalah Algoritma RC4 dan One Time Pad, Algoritma RC4 dan One Time Pad merupakan salah satu dari algoritma kunci. Sampai saat ini, algoritma RC4 dan One Time Pad masih dipercaya sebagai metode penyandian, kriptografi RC4 dan One Time Pad menggunakan kunci yang sama untuk enkripsi dan dekripsi.

METODE PENELITIAN

Dalam penelitian ini melakukan beberapa hal untuk mendapatkan data yang diperlukan [13], antara lain:

1. Metode Pengumpulan Data
   Beberapa metode pengumpulan data yang dilakukan yaitu:
   a. Studi kepustakaan *(library research)*
      Untuk mendapatkan hasil teori yang valid untuk dijadikan sebuah landasan, mencari beberapa buku referensi dari beberapa perpustakaan seperti mencari buku tentang keamanan data, kriptografi dan Algoritma RC4 dan One Time Pad.
   b. Pengumpulan data melalui surfing *(field research)*
      Pencarian atau penjelajahan untuk mencari data yang dapat dijadikan landasan yang sesuai melalui internet, seperti mencari file artikel yang membahas masalah kriptografi, keamanan data dan Algoritma RC4 dan One Time Pad.
   c. Wawancara (interview)
      Melakukan konsultasi atau tanya jawab secara langsung kepada orang yang lebih mengetahui tentang kriptografi dan Algoritma RC4 dan One Time Pad yang dibahas.
2. Metode Perancangan Sistem
   a. Analisis Kebutuhan
      Analisis kebutuhan adalah yaitu analisa Algoritma RC4 dan One Time Pad yang dilakukan untuk menentukan input dan output yang diinginkan berdasarkan rumus yang ada.
   b. Analisa dan Perancangan Sistem
      Perancangan sistem merupakan tahapan yang dilakukan untuk membuat sebuah rancangan program berdasarkan input dan output yang diinginkan.
   c. Implementasi Sistem
      Setelah pembuatan perancangan sistem maka langkah selanjutnya adalah mengimplementasi hasil perancangan kedalam program.
   d. Evaluasi Sistem
      Evaluasi merupakan langkah setelah Algoritma RC4 dan One Time Pad dimplementasikan untuk mengetahui kesalahan atau trouble yang mungkin terjadi, sampai dipastikan sistem dapat berjalan dengan sempurna.
   e. Penulisan laporan penelitian

HASIL DAN PEMBAHASAN

3.1. Analisa Keamanan

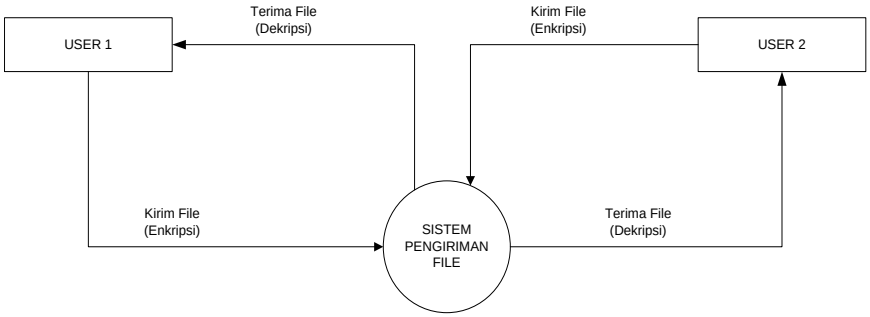
Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Salah satu hal yang penting dalam komunikasi menggunakan komputer dan dalam jaringan komputer untuk menjamin keamanan pesan, data, ataupun informasi adalah enkripsi. Disini enkripsi dapat diartikan sebagai kode atau chipper. Sebuah sistem pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk kata dari informasi atau yang merupakan bagian dari pesan, data, atau informasi yang dikirim. Sebuah chipper menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari suatu pesan asli (plainteks) menjadi *cryptogram* yang tidak dimengerti. Karena sistem chipper merupakan suatu sistem yang telah siap untuk diautomasi, maka teknik ini digunakan dalam sistem keamanan jaringan computer [14].

3.2. Proses Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu[15]:

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-repudiasi, atau penyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

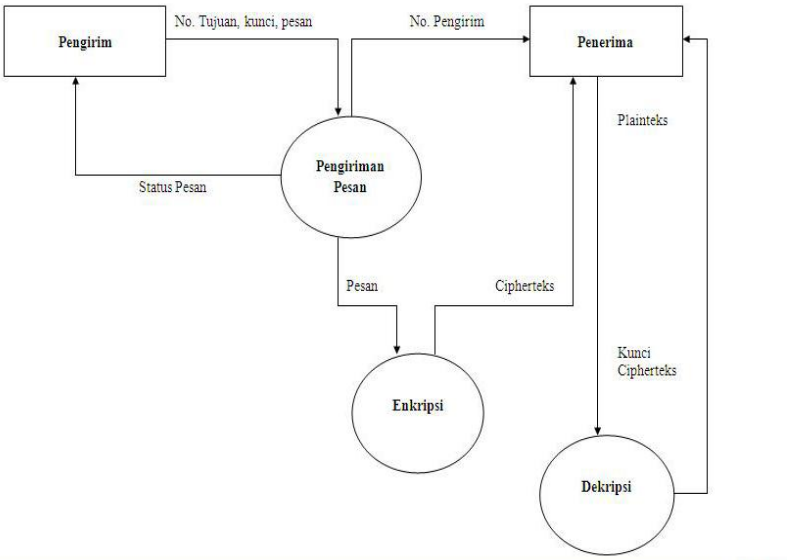


Gambar 1. DFD Proses Enkripsi dan Dekripsi

Keterangan:

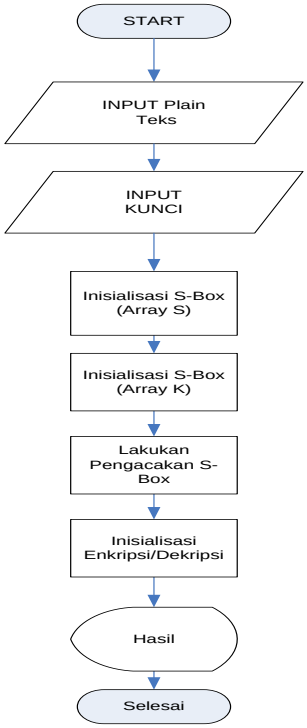
1. Pengiriman akan mengirim file kepada penerima dengan mengiputkan kirim file, lalu sistem akan memproses plainteks menjadi cipherteks sesuai yang dibuat.
2. Setelah file di enkripsi maka sistem akan mengirimkan cipherteks kepada sipenerima beserta dengan terima file.
3. Setelah pesan tersebut dikirim, maka sipengirim dapat melihat status file terkirim atau tidak.
4. Untuk membaca file tersebut, maka sipenerima harus mendekripsikannya terlebih dahulu dengan menginputkan kunci dan cipherteks tersebut, setelah itu maka tampillah file yang sebenarnya yang disebut dengan plainteks.

Dari Context Diagram yang diperlihatkan pada gambar 1, dapat dibuat diagram alir data (Data Flow Diagram) yang merupakan penjelasan lebih rinci dari DCD pada setiap tahapan. DFD dapat dilihat pada gambar 2 di bawah ini.



Gambar 2. DFD level 0 Proses Enkripsi dan Dekripsi

Untuk menerangkan proses enkripsi/dekripsi RC4 dalam bentuk flowchart dapat dilihat pada gambar 3 di bawah ini:



Gambar 3. Proses Enkripsi/Dekripsi RC4

Gambar 3 merupakan proses enkripsi dan dekripsi RC4 dapat dijelaskan sebagai berikut, setelah start atau mulai maka selanjutnya masukkan plainteks atau pilih file yang akan dienkripsi, selanjutnya masukkan kunci untuk mengenkripsi/mendekripsi file. Setelah menentukan kunci, maka sistem akan membaca atau mengenali dan mengubah kata di S-Box ke Array S dan membaca atau mengenali dan mengubah S-Box ke Array K. Setelah membaca dan mengenali maka sistem akan melakukan pengacakan data di S-Box. Setelah pengacakan, maka sistem akan mengenkripsi atau mendekripsi plainteks dan menampilkan hasil enkripsi/dekripsi, selesai.

3.3. Enkripsi RC4

Untuk menunjukkan bagaimana RC4 bekerja pada tingkat dasar, state-array 4 bit. Hal ini dikarenakan akan sangat sulit menggambarkan proses RC4 secara manual dengan state-array 256 bit. Kali ini akan dienkripsi kata HALO dengan kunci 2573. Pertama, menginisialisasi array S 4 bit sehingga terbentuk state-array S dan state-array K sebagai berikut.

Array S	0	1	2	3
Array K	2	5	7	3

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

Iterasi 1

i = 0

j = (0 + S[0] + K [0 mod 4]) mod 4

j = (j + S[i] = K[(i Mod 4)] Mod 4

= (0 + 0 + 2) mod 4 = 2

Swap (S[0],S[2])

Sehingga hasil array S adalah

Array S	2	1	0	3

Iterasi 2

i = 1

j = (2 + S[1] + K [1 mod 4]) mod 4

= (2 + 1 + 5) mod 4 =0

Swap (S[1],S[0])

Sehingga hasil array S adalah

Array S	1	2	0	3

Iterasi 3

i = 2

j = (0 + S[2] + K [2 mod 4]) mod 4

= (0 + 0 + 7) mod 4 = 3

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

Sehingga hasil array S adalah

Array S	1	2	3	0

Iterasi 4

i = 3
j = (3 + S[3] + K [3 mod 4]) mod 4
= (3 + 0 + 3) mod 4 = 2
Swap (S[3],S[2])

Sehingga hasil array S adalah

Array S	1	2	0	3

Setelah melakukan KSA, akan dilakukan *Pseudo Random Generation Algorithm* (PRGA). PRGA akan dilakukan sebanyak 4 kali dikarenakan plainteks yang akan dienkripsi berjumlah 4 karakter. Hal ini disebakan karena dibutuhkan 1 kunci dan 1 kali pengoperasian XOR untuk tiap-tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

Array S	1	2	0	3

Inisialisasi

i = 0
j = 0

Iterasi 1

i = (0 + 1) mod 4 = 1
j = (0 + S[1]) mod 4
= (0+ 2) mod4 = 2
swap (S[1],S[2])

1	0	2	3

K1 = S[(S[1]+S[2]) mod 4]
= S[2mod 4]
= S[2]
= 2
K1 = 00000010

Iterasi 2

i = (1 + 1) mod 4 = 2
j = (2 + S[2]) mod 4
= (2+ 2) mod4 = 0
swap (S[2],S[0])

2	0	1	3

K2 = S[(S[2]+S[0]) mod 4]
= S[3mod 4]
= 3
K2 = 00000011

Iterasi 3

i = (2 + 1) mod 4 = 3
j = (0 + S[3]) mod 4 = (0+ 3) mod4 = 3
swap (S[3],S[3])

1	0	2	3

K3 = S[(S[3]+S[3]) mod 4]
= S[6mod 4] = 2
K3 = 00000010

Iterasi 4

i = (3 + 1) mod 4 = 0
j = (3 + S[0]) mod 4
= (3+ 1) mod4 = 0
swap (S[0],S[0])

1	0	2	3

K1 = S[(S[0]+S[0]) mod 4]
= S[2mod 4] = 2
K1 = 00000010

Setelah menemukan kunci untuk setiap karakter, maka dilakukan operasi XOR antara karakter pada plaintext dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada plainteks yang digunakan dapat dilihat pada tabel 1 di bawah ini.

Tabel 1. Kode ASCII untuk setiap karakter plainteks yang digunakan

HURUF	KODE ASCII (8 Bit)
H	01001000
A	01000001

L	01001100
O	01001111

Proses XOR dari kunci bisa dilihat pada tabel 2 di bawah ini

Tabel 2. Proses XOR Plainteks dan kunci

	H	**A**	**L**	**O**
Plainteks	01001000	01000001	01001100	01001111
Key	00000010	00000011	00000010	00000010
Cipherteks	01001010 (L)	01000010 (B)	01001110 (N)	01001101 (M)

Dalam Proses pendekripsian dilakukan dengan proses XOR antara kunci dekripsi yang sama dengan kunci dekripsi dengan cipherteks yang dapat dilihat pada tabel 3 berikut:

Tabel 3. Proses XOR kunci dekripsi dengan Hipherteks

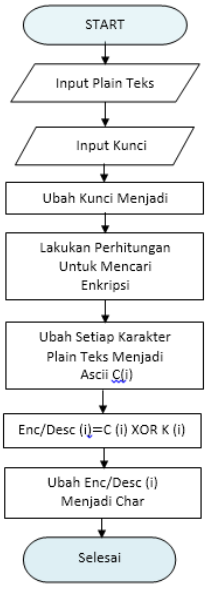
Cipherteks	01001010 (L)	01000010 (B)	01001110 (N)	01001101 (M)
Key	00000010	00000011	00000010	00000010
Plainteks	01001000 (H)	01000001 (E)	01001100 (L)	01001111 (O)

3.4. Flowchart Proses Enkripsi/Dekripsi One Time Pad (OTP)

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi, data disandikan (encrypted) dengan mengunakan sebuah kunci (key). Untuk membuka (men-decrypt) data tersebut, juga digunakan kunci yang dapat sama dengan kunci untuk mengenkripsi (privat key) atau dengan kunci yang berbeda (public key).

Keamanan dari enkripsi bergantung pada beberapa faktor. Pertama, algoritma enkripsi harus cukup kuat sehingga sulit untuk men-decrypt cipherteks dengan dasar cipherteks tersebut. Lebih jauh lagi, keamanan dari algoritma enkripsi bergantung pada kerahasiaan dari kuncinya bukan algoritmanya, yaitu dengan asumsi bahwa adalah sangat tidak praktis untuk men-deycrypt informasi dengan dasar chiperteks dan pengetahuan tentang algoritma dekripsi atau enkripsi. Atau dengan kata lain, tidak perlu menjaga kerahasiaan dari algoritma tetapi cukup dengan kerahasiaan kuncinya.

Dekripsi digunakan untuk mengembalikan data-data atau informasi sehingga dapat dibaca oleh orang yang berhak. Dengan dekripsi, data dikembalikan kedalam bentuk semula sehingga dapat dibaca dengan baik. Gambar 4 menerangkan proses enkripsi/dekripsi One Time Pad dalam bentuk flowchart dapat dilihat pada gambar 4 di bawah ini:



Gambar 4. Proses Enkripsi/Dekripsi One Time Pad

Gambar 4 merupakan proses enkripsi dan dekripsi One Time Pad dapat dijelaskan sebagai berikut. Setelah start, selanjutnya user menginput plainteks, setelah itu maka masukkan kata kunci. Selanjutnya ubah setiap kata kunci ke dalam kode Ascii dan disimpan dan ditampung dalam K(i). Setelah mengubah kata kunci ke dalam Ascii, maka selanjutnya sistem akan mengubah setiap karakter plainteks kedalam Ascii dan ditampung dalam C(i). Setelah itu dilakukan perhitungan untuk mengetahui enkripsi setiap karakter.

Setelah perhitungan, selanjutnya kata kunci dan plainteks akan di enkripsi dan hasil enkripsi akan ditampung dalam Enc/Desc (i). Selanjutnya sistem akan mengubah hasil Enc/Desc (i) kedalam karakter, selesai.

3.5 Enkripsi One Time Pad (OTP)

Contohnya adalah pada saat akan mengirimkan file terhadap seseorang, file tersebut pastinya bersifat rahasia. Pada pembahasan ini One Time Pad akan mengenkripsi file sehingga file tersebut aman. Dibawah ini akan di jelaskan contoh penggunaan algoritma One Time Pad pada sebuah file. Sebagai contoh , Sebuah file "**UNIKA**" akan dienkripsi dengan kunci "**MEDAN**" dengan perhitungan sebagai berikut, maka akan diperoleh hasil dapat dilihat pada tabel 4 di bawah ini.

Tabel 4 Ascii File

Plain Teks	Ascii
U	85
N	78
I	73
K	75
A	65

Pada tabel ascii kunci untuk menentukan plainteksnya dapat dilihat pada tabel 5 berikut:

Tabel 5. Ascii Kunci

Teks Kunci	Ascii
M	77
E	69
D	68
A	65
N	78

Dari tabel 4 dan table 5 dapat dilakukan proses perhitungan enkripsi dapat dilihat pada tabel 6.

Tabel 6. Proses Perhitungan Enkripsi

Plainteks	**85**	**78**	**73**	**75**	**65**
Kunci	**77**	**69**	**68**	**65**	**78**
Plainteks XOR Kunci	**24**	**11**	**13**	**10**	**15**

Jadi hasilnya adalah

Desimal	**24**	**11**	**13**	**10**	**15**
Char	**CAN**	**VT**	**CR**	**LF**	**SI**

Untuk mendeskripsinya, maka dilakukan proses kebalikannya dapat dilihat pada tabel 7 di bawah ini.

Tabel 7. Ascii Enkripsi

Plain Teks/Char	Desimal
CAN	24
VT	11
CR	13
LF	10
SI	15

Tabel ascii kunci enkripsi untuk menghasilkan file enkripsi dapat dilihat pada tabel 8 di bawah ini.

Tabel 8. Ascii Kunci Enkripsi

Teks Kunci	Desimal
M	77
E	69
D	68
A	65
N	78

Jadi hasilnya adalah

Chiperteks	**24**	**11**	**13**	**10**	**15**
Kunci	**77**	**69**	**68**	**65**	**78**
File XOR Kunci	**85**	**78**	**73**	**75**	**65**

Dari tabel diatas dapat disimpulkan sebagai berikut:

File Enkripsi **XOR** Kunci : **85 – 78 – 73 – 75 – 65**

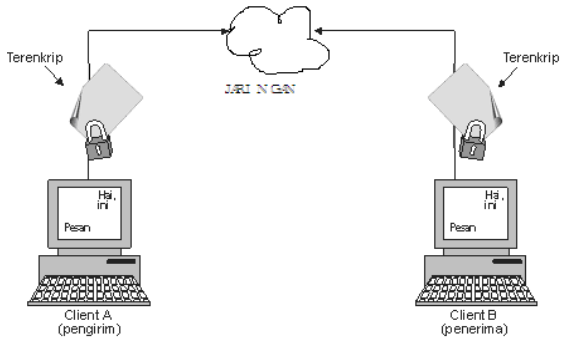
Lalu angka desimal tersebut di konversi kembali ke bentuk char, sehingga hasilnya seperti di bawah ini:

= **U - N - I - K - A**

3.6. Cara Kerja Enkripsi Dalam Jaringan

Proses enkripsi informasi asal yang dapat dimengerti disimbolkan oleh plain teks, yang kemudian oleh algoritma enkripsi diterjemahkan menjadi informasi yang tidak dapat dimengerti yang disimbolkan dengan Cipherteks. Proses enkripsi terdiri dua

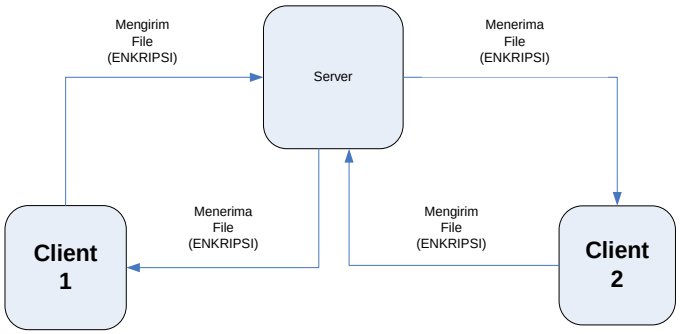
yaitu algoritma dan kunci. Kunci biasanya merupakan suatu string bit yang pendek yang mengendalikan algoritma. Algoritma enkripsi akan menghasilkan hasil yang berbeda bergantung pada kunci yang digunakan. Mengubah kunci dari enkripsi akan mengubah output dari algoritma enkripsi. Sekali cipherteks telah dihasilkan, kemudian ditransmisikan. Pada bagian penerima selanjutnya cipherteks yang diterima diubah kembali ke plainteks dengan algoritma dan kunci yang sama. Adapun gambar dari proses enkripsi dan dekripsi dalam sebuah jaringan dapat dilihat pada gambar 5 di bawah ini.



Gambar 5. Arsitektur Struktur yang akan dirancang

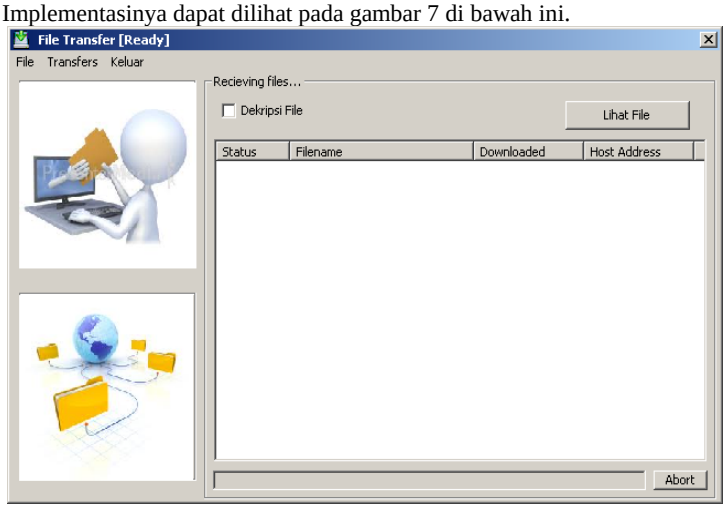
3.7. Hubungan Antar *Client*

Program ini digunakan untuk komunikasi antar komputer, umumnya lewat network atau internet. *Server* berfungsi sebagai pemberi informasi pada setiap *client* yang meminta layanan tersebut dan mengirimkannya ke setiap *client* lainnya dan jalur pada sistem, dapat dilihat pada gambar 6 di bawah ini.



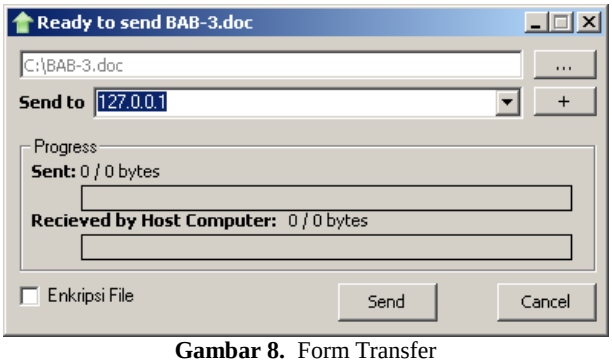
Gambar 6. *Server* yang memberikan RESPONSE ke *Client*

Form ini akan tampil setelah *user* menekan tombol Waiting pada Halaman Utama seperti pada Gambar 7 untuk melihat Menu yang di inginkan oleh *user*. Implementasinya dapat dilihat pada gambar 7 di bawah ini.



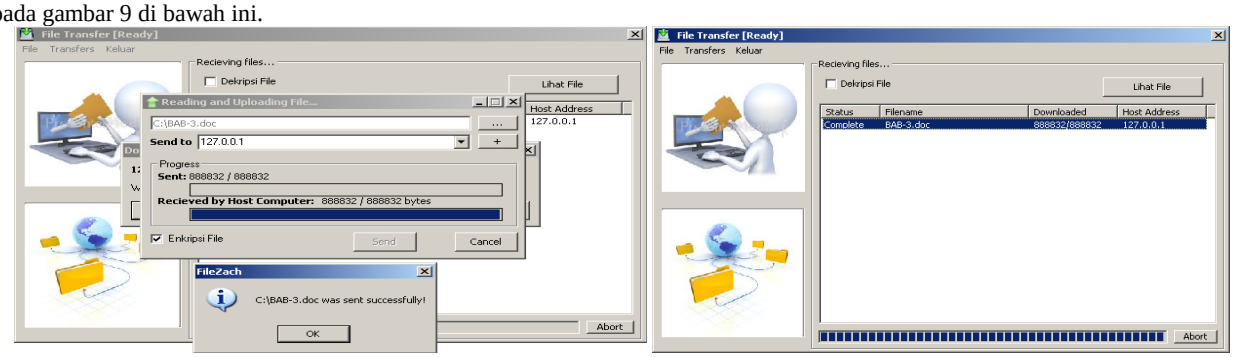
Gambar 7. Form Menu Utama

Form Transfer ini akan tampil setelah *user* memilih Menu Transfer pada Menu Pilihan pada gambar 8 yang berfungsi untuk transfer file, jika *user* ingin mengirim file maka *user* harus menginputkan select a File, Ip.Addres, Send. Implementasinya dapat dilihat pada gambar 8 di bawah ini.



Gambar 8. Form Transfer

Form ini akan tampil pada saat memilih Proses Kirim File pada Menu Utama pada Gambar 9 yang berfungsi untuk proses pengiriman file yang telah terenkripsi dan untuk menampilkan file yang diterima dari client. Implementasinya dapat dilihat dilihat pada gambar 9 di bawah ini.



Gambar 9. Form Proses Kirim dan terima File

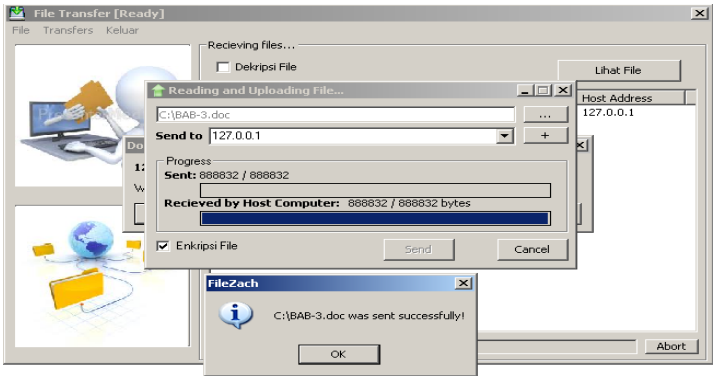
Materi yang akan diujikan pada aplikasi Transfer File ini adalah sebagai berikut:

1. Pengiriman File

Proses ini berfungsi untuk mengenkripsi file dan mengirimkan file ke client. Langkah-langkah yang dilakukan user dalam proses ini adalah sebagai berikut:

a. Input select a file
   User diminta untuk menginputkan select a file kemana pesan akan dikirimkan.

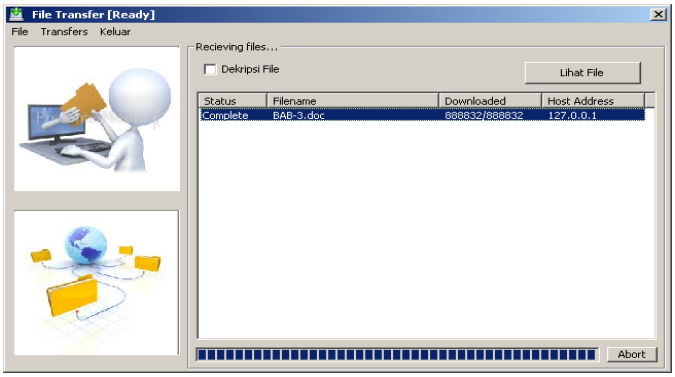
b. Input sent to
   Sent to yang diinputkan merupakan Ip.Addres yang sudah di setting terlebih dahulu dari server dan client. Proses selengkapnya dapat dilihat pada gambar 11. Dan user diminta untuk menekan tombol send, lalu file tersebut akan dikirimkan ke komputer client atau dari komputer client ke komputer server. Proses selengkapnya dapat dilihat pada gambar 11 di bawah ini.



Gambar 11. Proses Pengiriman File

2. Penerimaan File

File yang dikirim dari komputer server ke client ataupun sebaliknya dari komputer client ke server dapat di lihat dari download tempat program itu disimpan. Proses selengkapnya dapat dilihat pada gambar 12 di bawah ini.



Gambar 11. Penerimaan File

KESIMPULAN

Berdasarkan analisa dan perancangan yang dilakukan, kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Program dibuat untuk mengirimkan sebuah file dari satu komputer ke komputer lainnya.
2. Dalam pengiriman file tersebut terlebih dahulu di enkripsi dan di dekripsi secara otomatis setelah sampai di tujuan.
3. Aplikasi ini dapat melakukan enkripsi dan dekripsi dengan menggunakan gabungan RC4 dan One Time Pad.
4. Perangkat Lunak ini hanya bisa melakukan enkripsi pada plain text.
5. Algoritma ini akan lebih tahan terhadap serangan karena menggunakan 2 (dua) enkripsi sekaligus.

DAFTAR PUSTAKA

[1] L. Sitorus, "Algoritma dan Pemrograman," *Andi Yogyakarta*, 2015. [Online]. Available: https://books.google.co.id/books?id=MRHwCgAAQBAJ&printsec=frontcover&dq=inauthor:%22Lamhot+Sitorus%22&hl=en&sa=X&ved=0ahUKEwjiz6iIjZfnAhVp7XMBHYf7C0QQ6AEIKjAA#v=onepage&q&f=false. [Accessed: 22-Jan-2020].

[2] T. Limbong *et al.*, "The implementation of computer based instruction model on Gost Algorithm Cryptography Learning," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 420, no. 1, p. 12094.

[3] D. Ariyus, "Kriptografi keamanan data dan komunikasi," *Yogyakarta Graha Ilmu*, 2006.

[4] M. T. Suryadi *et al.*, "SMS Security System on Mobile Devices Using Tiny Encryption Algorithm," *IOP Conf. Ser. J. Phys. Conf. Ser.*, vol. 1007, p. 12037, 2018, doi: 10.1088/1742-6596/1007/1/012037.

[5] R. N. Ibrahim, "PERANGKAT LUNAK KEAMANAN DATA MENGGUNAKAN ALGORITMA KRIPTOGRAFI SIMETRI TINY ENCRIPTION ALGORITHM (TEA)," *JURNAL COMPUTECH & BISNIS*, 2019. [Online]. Available: http://jurnal.stmik-mi.ac.id/index.php/jcb/article/view/191/215. [Accessed: 02-Feb-2020].

[6] H. Sahara, "Implementasi Pengamanan Pesan Chatting menggunakan Metode Vigenere Cipher dan Cipher Block Chainning," *MEANS (Media Inf. Anal. dan Sist.*, vol. 3, no. 2, pp. 173–178, 2018.

[7] S. B. Sinaga and Y. Hasan, "Pengukuran Kualitas Jaringan Internet Dengan Sinyal 3G Lte Pada STMIK Budi Darma Medan dengan Metode Quality Of Service (QoS)," vol. 2, no. 2, pp. 105–108, 2017.

[8] H. Saragih, G. Gusvita, B. Reza, D. Setiyadi, and R. Akbar, "Pengembangan Sistem Informasi Distribusi Informasi Sekolah Melalui Sms Gateway Dengan Zachman Framework," *J. Sist. Inf.*, vol. 8, no. 1, p. 32, 2013, doi: 10.21609/jsi.v8i1.320.

[9] M. D. Herminingtyas and R. Sholihah, "Implementasi algoritma rc4 untuk enkripsi keamanan data," pp. 1–11, 1987.

[10] Y. Ariyanto, "Algoritma Rc4 Dalam Proteksi Transmisi Dan Hasil Query Untuk Ordbms Postgresql," *J. Inform.*, vol. 10, no. 1, 2010, doi: 10.9744/informatika.10.1.53-59.

[11] N. E. Saragih, "IMPLEMENTASI ALGORITMA ONE TIME PAD PADA PESAN," *J. Ilm. Matrik*, vol. 20, no. 1, pp. 31–40, 2018.

[12] F. Diani and Y. Widhiyasana, "Enkripsi SMS dengan Menggunakan One Time Pad (OTP) dan Kompresi Lempel-Ziv-Welch (LZW)," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 3, pp. 3–8, 2018.

[13] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D.* Bandung: PT Alfabet, 2016.

[14] T. Limbong, "Pengujian Kriptografi Klasik Caesar Chipper Menggunakan Matlab," *no. Sept.*, vol. 2017, 2015.

[15] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.