ABSTRAK

MEMBANGUN APLIKASI KEAMANAN TRANSMISI DATA MULTIMEDIA MENGGUNAKAN KRIPTOGRAFI ALGORITMA *DATA ENCRYPTION STANDARD* (DES)

Tarbudi
10103075

Dengan perkembangan di bidang teknologi informasi, teknologi komunikasi data dan teknologi multimedia pada saat ini yang begitu pesat, maka hal yang sangat penting yang harus dipikirkan adalah masalah keamanan karena akan menjamin data khususnya data multimedia agar tidak dicuri oleh orang yang tidak berhak.

Untuk mengatasi hal tersebut, teknik keamanan yang dipakai adalah algoritma DES (*Data Encryption Standard*) yang berfungsi untuk menyediakan keamanan transmisi data multimedia dengan cara mengenkrip data multimedia sehingga bagi orang yang tidak berhak tidak akan dapat membaca data multimedia tersebut tanpa memiliki kuncinya, teknik ini sangat efektik karena dapat menjaga kerahasiaan data khususnya data multimedia dan juga memerlukan waktu yang sangat lama untuk dapat menemukan kunci yang benar.

Kata Kunci : DES, Kriptografi, Keamanan Data Multimedia

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini perkembangan dibidang teknologi informasi atau yang lebih dikenal dengan sebutan *Information Technology* (IT) sangat cepat dan pesat, dimana telah memberikan pengaruh yang besar pada berbagai aspek kehidupan manusia. Perkembangan teknologi informasi didukung dengan perkembangan pada *hardware* (perangkat keras) dan *software* (perangkat lunak) yang terus menerus menunjang kemudahan bagi banyak orang dalam memenuhi kebutuhan informasi.

Berkembangnya teknologi informasi secara tidak langsung berpengaruh terhadap bidang komunikasi data, dimana bidang komunikasi data ini menjadi jalur tranmisi bagi data informasi yang mengalir dan menjadi penghubung dari satu orang ke orang lain. Pada zaman sekarang ini banyak sekali aplikasi komunikasi data yang banyak digunakan orang sebagai sarana berkomunikasi, diantaranya adalah jaringan area lokal pada skala ruang lingkup yang kecil dan internet pada skala ruang lingkup yang luas.

Pada saat sekarang ini teknologi multimedia juga berkembang dengan pesat, sebagai contoh *video stream*ing dan *video on demand*, dimana dulu teknologi ini sulit untuk dinikmati oleh banyak orang dan hanya kalangan tertentu saja yang dapat menikmatinya, sekarang seiring berkembangnya teknologi komunikasi data, multimedia ini sudah tersebar luas dan dapat dinikmati oleh semua orang.

Dengan adanya perkembangan di bidang teknologi informasi, teknologi komunikasi data dan teknologi multimedia, maka hal yang sangat

penting yang harus dipikirkan adalah masalah keamanan. Masalah keamanan sangat penting karena akan menjamin data khususnya data multimedia agar tidak dicuri oleh orang yang tidak berhak. Salah satu teknik keamanan yang terkenal adalah kriptografi yaitu ilmu yang dapat menyandikan data sehingga kerahasiaan data tetap terjaga, enkripsi DES (*Data Encryption Standard*) adalah salah satu teknik algoritma kriptografi yang berfungsi untuk menyediakan keamanan data multimedia dengan cara mengenkrip data tersebut sehingga bagi orang yang tidak berhak tidak akan dapat membaca data tersebut tanpa memiliki kuncinya.

Berdasarkan latar belakang diatas maka pada pembutan tugas akhir ini diambil dengan judul "**MEMBANGUN APLIKASI KEAMANAN TRANSMISI DATA MULTIMEDIA MENGGUNAKAN KRIPTOGRAFI ALGORITMA *DATA ENCRYPTION STANDARD* (DES)".**

1.2 Identifikasi Masalah

Seperti yang dijelaskan diatas, terdapat perumusan masalah yang dapat dirumuskan yaitu sebagai berikut:

1. Bagaimana cara membangun aplikasi keamanan transmisi data multimedia dengan menggunakan kriptografi algoritma *Data Encryption Standard* (DES)?
2. Bagaimana mengintegrasikan protokol keamanan kedalam aplikasi ini?

1.3 Maksud dan Tujuan

Maksud dari penulisan tugas akhir ini adalah untuk membangun aplikasi keamanan transmisi data multimedia menggunakan kriptografi algoritma *Data Encryption Standard* (DES).

Tujuan yang akan dicapai dalam penelitian ini adalah

1. Untuk mengamankan transmisi data multimedia.
2. Pengiriman file yang dienkripsi tidak menjadi beban bagi pengguna.

1.4 Batasan Masalah

Agar aplikasi yang dihasilkan tidak terlalu jauh dari tujuannya. Maka dalam penelitian ini, penulis membatasi masalah sebagai berikut :

1. Program aplikasi yang akan dibangun akan di implementasi pada jaringan LAN (*Local Area Network*).
2. Program aplikasi ini akan dijalankan pada teknik komunikasi *client-server*.
3. Implementasi pada aplikasi ini hanya menggunakan satu client.
4. Algoritma kriptografi yang digunakan adalah *Data Encryption Standard* (DES).
5. Didalam aplikasi yang akan dibangun ini tidak terdapat proses kompresi data.
6. File video yang digunakan pada simulasi yaitu file berformat mpg, wmv, dan avi.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut :

1. Tahap pengumpulan data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah :
Studi Literatur, yaitu pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian, dan juga melakukan pengumpulan data dari berbagai sumber di internet yang tujuanya untuk mengetahui perkembangan terbaru dari data yang diambil sebagai bahan dalam pembuatan tugas akhir.

2. Tahap pembuatan perangkat lunak.

Teknik analisis data dalam pembuatan perangkat lunak menggunakan paradigma perangkat lunak secara *waterfall*, yang digambarkan dibawah ini:

BAB 2
LANDASAN TEORI

2.1 Kriptografi

2.3.1 Sejarah Kriptografi

Kriptografi atau yang sering dikenal dengan sebutan ilmu penyandian data, adalah suatu bidang ilmu dan seni (*art and science*) yang bertujuan untuk menjaga kerahasiaan suatu pesan yang berupa data data dari akses oleh orang-orang atau pihak-pihak lain yang tidak berhak sehingga tidak menimbulkan kerugian. Bidang ilmu Kriptografi ini semula hanya populer dibidang militer dan bidang intelijen untuk menyandikan pesan-pesan panglima perang kepada pasukan yang berada di garis depan akan tetapi seiring dengan semakin berkembangnya teknologi terutamanya teknologi informasi dan semakin padatnya lalu lintas informasi yang terjadi tentu saja semakin menuntut adanya suatu komunikasi data yang aman, bidang ilmu ini menjadi semakin penting. Sekarang bidang ilmu ini menjadi salah satu isu suatu topik riset yang tidak habis-habisnya diteliti dengan melibatkan banyak peneliti.

Ilmu kriptografi sebenarnya sudah mulai dipelajari manusia sejak tahun 400 SM, yaitu pada zaman Yunani kuno. Dari catatan bahwa “Penyandian Transposisi” merupakan sistem kriptografi pertama yang digunakan atau dimanfaatkan. Bidang ilmu ini terus berkembang seiring dengan kemajuan peradaban manusia, dan memegang peranan penting dalam strategi peperangan yang terjadi dalam sejarah manusia, mulai dari sistem kriptografi “*Caesar Chiper*” yang terkenal pada zaman Romawi kuno, “*Playfair Cipher*” yang digunakan Inggris dan “*ADFVGX Cipher*” yang digunakan Jerman pada Perang Dunia I, hingga algoritma-algoritma kriptografi rotor yang populer pada Perang Dunia II, seperti Sigaba / M-134 (Amerika Serikat), Typex (Inggris), Purple (Jepang), dan mesin kriptografi legendaris Enigma (Jerman).

2.3.2 Tujuan Kriptografi

Dalam teknologi informasi, telah dan sedang dikembangkan cara-cara untuk menangkal berbagai bentuk serangan semacam penyadapan dan pengubahan data yang dikirimkan. Salah satu cara yang ditempuh mengatasi masalah ini ialah 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 (*authenticatioan*). Privasi mengandung arti bahwa data yang diinginkan hanya dapat dimengerti informasinya oleh penerima yang sah atau berhak. Sedangkan keotentikan mencegah pihak ketiga untuk mengirimkan data yang salah atau mengubah data yang dikirimkan .

Adapun tujuan sistem kriptografi adalah sebagai berikut:

1. *Convidentiality*

Yaitu memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi.

2. *Massage Integrity*

Yaitu memberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat data dibuat atau dikirim sampai dengan saat data tersebut dibuka.

3. *Non-repudiation*

Yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut.

4. *Authentication*

Yaitu memberikan dua layanan. Pertama mengidentifikasikan keaslian suatu pesan dan memberikan jaminan keotentikannya. Kedua untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem.

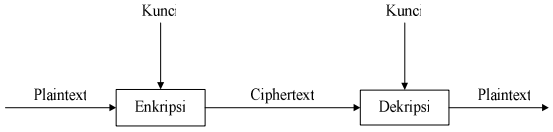
2.3.3 Algoritma Kriptografi

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu :

1. Kunci Simetris

Kunci Simetris adalah jenis kriptografi yang paling umum digunakan. Kunci untuk membuat pesan yang di sandikan sama dengan kunci untuk membuka pesan yang disandikan itu. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama persis. Siapapun yang memiliki kunci tersebut termasuk pihak-pihak yang tidak diinginkan dapat membuat dan membongkar rahasia ciphertext. Kelompok algoritma simetris adalah OTP, DES, RC2, RC4, RC5, RC6, IDEA, Twofish, Magenta, FEAL, SAFER, LOKI, CAST, Rijndael (AES), Blowfish, GOST, A5, Kasumi dan lain-lain.

Proses enkripsi-deskripsi algoritma kriptografi kunci simetris dapat dilihat pada gambar dibawah ini :



Gambar 2. 1 Proses enkripsi-deskripsi kunci simetris

Algoritma kriptografi simeteris dibagi menajdi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data (per blok).

Kelebihan algoritma simetris ini adalah kecepatan proses enkripsi dan deskripsinya yang jauh lebih cepat dibandingkan dengan algoritma asimetris. Sedangkan kelemahan algoritma ini adalah permasalahan distribusi kunci (*key distribution*). Seperti yang telah dibahas, proses enkripsi dan deskripsi menggunakan kunci yang sama. Sehingga muncul persoalan menjaga kerahasian kunci, yaitu pada saat pengiriman kunci pada media yang tidak aman seperti internet. Tentunya jika kunci ini sampai hilang atau sudah dapat ditebak oleh orang lain (orang yang tidak berhak), maka kriptosistem ini sudah tidak aman lagi.

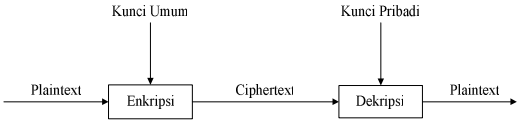
Kelemahan lain adalah masalah efisiensi jumlah kunci. Jika terdapat *n* user, maka diperlukan $n(n-1)/2$ kunci, sehingga untuk jumlah user yang sangat banyak, sistem ini tidak efisien lagi.

2. Kunci Asimetris

Kunci asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang satu lagi untuk deskripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsikan suatu pesan, data ataupun informasi, sedangkan hanya satu orang saja yang memiliki rahasia

tertentu dalam hal ini kunci privat untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA.

Proses enkripsi-deskripsi algoritma kunci asimetris dapat dilihat pada gambar dibawah ini :



Gambar 2. 2 Proses enkripsi-deskripsi kunci asimetris

Pada algoritma *public key* ini, semua orang dapat mengenkripsi data dengan memakai *public key* penerima yang telah diketahui secara umum. Akan tetapi data yang telah terenkripsi tersebut hanya dapat dideskripsi dengan menggunakan *private key* yang hanya diketahui oleh penerima.

Keuntungan utama dari algoritma ini adalah memberikan jaminan keamanan kepada siapa saja yang melakukan pertukaran informasi meskipun diantara mereka tidak ada kesepakatan mengenai keamanan data terlebih dahulu maupun saling tidak mengenal satu sama lainnya. Beberapa contoh konsep yang menggunakan algoritma ini adalah skema enkripsi Elgamal, RSA, Diffie–Hellman (DH), DSA (*Digital Signature Algorithm*) dan lain-lain

2.3.4 Aritmetika Modular

Aritmetika modular merupakan operasi matematika yang banyak diimplementasikan pada metode kriptografi. Pada metoda kriptografi simetris, operasi aritmetika modular yang sering dipakai adalah operasi penjumlahan modulo dua dan operasi XOR (Exclusive OR) dengan simbol $\oplus$. Operasi modulo dua ini melibatkan bilangan 0 dan 1 saja sehingga identik dengan bit pada komputer. Seluruh kemungkinan nilai operasi XOR ini dapat dilihat pada table dibawah ini :

Tabel 2. 1 Operasi XOR

a	b	a $\oplus$ b
0	0	0
0	1	1
1	0	1
1	1	0

Dari tabel diatas dapat dilihat sifat-sifat unik operasi XOR yaitu : $A \oplus A=0$, $A \oplus 0= A$, $A \oplus 1 = A'$, dengan A' adalah komplemen dari A.

2.3.5 Proses padding

Proses padding adalah proses penambahan byte-byte dummy berupa karakter NULL pada byte-byte sisa yang masih kosong pada blok terakhir plaintext, sehingga ukurannya menjadi sama dengan ukuran blok penyandian. Byte terakhir kemudian diisi dengan suatu informasi mengenai ukuran file pada blok terkahir.

2.2 Data Encryption Standard (DES)

2.4.1 Sejarah DES

DES (*Data Encryption Standard*) merupakan nama dari sebuah algoritma untuk mengenkripsi data yang dikeluarkan oleh *Federal Information Processing Standard* (FIPS) 46-1 Amerika Serikat. Algoritma dasarnya dikembangkan oleh IBM, NSA, dan NBS yang berperan penting dalam pengembangan bagian akhir algoritmanya. DEA dan DES telah dipelajari secara ekstensif sejak publikasi pertamanya, dan diketahui sebagai algoritma simetris yang paling baik dan paling banyak digunakan di dunia.

DES memiliki blok kunci 64 bit tetapi yang digunakan dalam proses eksekusi adalah 56 bit. Pada awalnya dirancang untuk implementasi secara

hardware. Penggunaan dalam sistem komunikasi mengharuskan pengirim dan penerima memiliki kunci rahasia yang sama, yang dapat digunakan untuk mengenkripsi dan mendeskripsi data yang dikirim atau diterima. DES juga dapat digunakan untuk enkripsi data-data pribadi dalam *harddisk.*

2.4.2 Tinjauan Umum

DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit

Skema global dari algoritma DES adalah sebagai berikut (lihat Gambar 2.9):

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok cipherteks.

Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal K_i. Keluaran dai fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya.

BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Seperti yang telah di jelaskan pada bab sebelumnya bahwa masalah yang muncul adalah bagaimana mengintegrasikan sistem keamanan data multimedia dengan cara mengenkripsi data multimedia tersebut sehingga dengan adanya sistem keamanan tersebut sedikitnya tidak mengganggu jalannya transmisi data multimedia antara client dan server.

Mungkin bagi sebagian orang dengan adanya sistem keamanan ini akan menjadi beban karena pasti sedikitnya akan mempengaruhi kinerja dari transmisi ini dan juga ukuran data multimedia setelah dienkripsi pasti membengkak, tetapi melihat pentingnya data multimedia yang ditransmisikan antar client-server maka penggunaan sistem keamanan dengan cara mengenkripsi data multimedia ini menjadi pertimbangan tersendiri, apalagi data yang ditransmisikan adalah data yang sangat rahasia sehingga client yang tidak berhak yang ingin mengetahui data apa yang sedang ditransmisikan dari server, maka client tersebut tidak akan mengetahuinya.

3.2 Analisis Kebutuhan Sistem

Perangkat keras (*Hardware*) yang digunakan sebagai implementasi atau simulasi dari program yang akan dibangun yaitu seperangkat komputer dengan spesifikasi sebagai berikut:

Komputer server:

1. Processor Intel Pentium IV 1.8 Ghz
2. Harddisk 40 GB
3. Monitor VGA
4. RAM 512 MB
5. DVD Rom
6. VGA 64 MB
7. Keyboard dan Mouse
8. Lancard SiS 900-Based PCI Fast Ethernet Adapter

Komputer Client:

1. Processor Intel Pentium IV 1.8 Ghz
2. Harddisk 40 GB
3. Monitor VGA
4. RAM 128
5. VGA 64 MB
6. Intel® Integrated Audio
7. Speaker
8. Keyboard dan Mouse
9. Lancard Realtek RTL8139 Family PCI Fast Ethernet NIC

3.3 Algoritma DES

Secara umum, algoritma DES terbagi menjadi 3 kelompok di mana kelompok yang satu dengan yang lain saling berintraksi dan terkait antara satu dengan yang lain. Kelompok-kelompok tersebut adalah: Pemrosesan kunci, enkripsi data 64 bit, dan deskripsi data 64 bit.

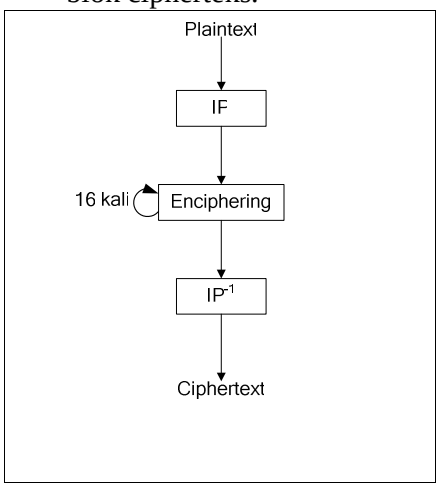
Algoritma DES dirancang untuk menulis dan membaca berita blok data yang terdiri dari 64 bit dibawah kontrol kunci 64 bit. Dalam pembacaan berita harus dikerjakan dengan menggunakan kunci yang sama dengan waktu menulis berita, dengan penjadwalan alamat kunci bit yang diubah sehingga proses membaca adalah kebalikan dari proses menulis.

Sebuah blok ditulis dan ditujukan pada permutasi dengan inisial IP, kemudian melewati perhitungan dan perhitungan tersebut sangat tergantung pada kunci kompleks dan pada akhirnya melewati permutasi yang invers dari permutasi dengan inisial IP^{-1}.

DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit.

Skema global dari algoritma DES adalah sebagai berikut (lihat Gambar 3.1):

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering*- sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok cipherteks.



Gambar 3. 1 Skema Global Algoritma DES

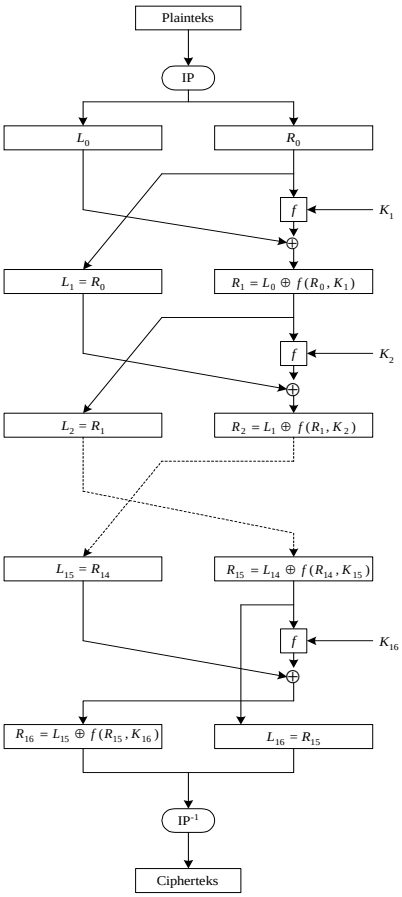
Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masing-

masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal K_i. Keluaran dai fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES.

Secara matematis, satu putaran DES dinyatakan sebagai:

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$



Gambar 3. 2 Algoritma Enkripsi DES

Perlu dicatat dari Gambar 3.2 bahwa jika (L_{16}, R_{16}) merupakan keluaran dari putaran ke-16, maka (R_{16}, L_{16}) merupakan pra-cipherteks (*pre-ciphertext*) dari *enciphering* ini. Cipherteks yang sebenarnya diperoleh dengan melakukan permutasi awal balikan, IP^{-1}, terhadap blok pra-cipherteks.

3.3.1 Permutasi Awal

Sebelum putaran pertama, terhadap blok plainteks dilakukan permutasi awal (*initial permutation* atau IP). Tujuan permutasi awal adalah mengacak plainteks sehingga urutan bit-bit di dalamnya berubah. Pengacakan dilakukan dengan menggunakan matriks permutasi awal berikut ini:

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Cara membaca tabel/matriks di atas: dua *entry* ujung kiri atas (58 dan 50) berarti:
"pindahkan bit ke-58 ke posisi bit 1"
"pindahkan bit ke-50 ke posisi bit 2", dst

3.3.2 Pembangkitan Kunci Internal

Karena ada 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K_1, K_2, …, K_{16}. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter.

Misalkan kunci eksternal yang tersusun dari 64 bit adalah *K*. Kunci eksternal ini menjadi masukan untuk permutasi dengan menggunakan matriks permutasi kompresi PC-1 sebagai berikut:

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Dalam permutasi ini, tiap bit kedelapan (*parity bit*) dari delapan *byte* kunci diabaikan. Hasil permutasinya adalah sepanjang 56 bit, sehingga dapat dikatakan panjang kunci DES adalah 56 bit. Selanjutnya, 56 bit ini dibagi menjadi 2 bagian, kiri dan kanan, yang masing-masing panjangnya 28 bit, yang masing-masing disimpan di dalam C_0 dan D_0:

C_0: berisi bit-bit dari K pada posisi
57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18, 10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36

D_0: berisi bit-bit dari K pada posisi
63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22, 14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4

Selanjutnya, kedua bagian digeser ke kiri (*left shift*) sepanjang satu atau dua bit bergantung pada tiap putaran. Operasi pergeseran bersifat *wrapping* atau *round-shift*. Jumlah pergeseran pada setiap putaran ditunjukkan pada Tabel 3.1 sebagai berikut:

Table 3. 1 Jumlah pergeseran bit pada setiap putaran

Putaran, i	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Misalkan (C_i, D_i) menyatakan penggabungan C_i dan D_i. (C_{i+1}, D_{i+1}) diperoleh dengan menggeser C_i dan D_i satu atau dua bit. Setelah pergeseran bit, (C_i, D_i) mengalami permutasi kompresi dengan menggunakan matriks PC-2 berikut:

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Dengan permutasi ini, kunci internal K_i diturunkan dari (C_i, D_i) yang dalam hal ini K_i merupakan penggabungan bit-bit C_i pada posisi:

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10, 23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2

dengan bit-bit D_i pada posisi:

41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48, 44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32

Jadi, setiap kunci internal K_i mempunyai panjang 48 bit.

Proses pembangkitan kunci-kunci internal ditunjukkan pada Gambar 3.3. bila jumlah pergeseran bit-bit pada Tabel 3.1 dijumlahkan semuanya, maka jumlah seluruhnya sama dengan 28, yang sama dengan jumlah bit pada C_i dan D_i. Karena itu, setelah putaran ke-16 akan didapatkan kembali $C_{16} = C_0$ dan $D_{16} = D_0$.

3.3.3 Enkripsi (*Enciphering*)

Proses *enciphering* terhadap blok plainteks dilakukan setelah permutasi awal. Setiap blok plainteks mengalami 16 kali putaran *enciphering*. Setiap putaran *enciphering* merupakan jaringan Feistel yang secara matematis dinyatakan sebagai

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi sebagai berikut:

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

Selanjutnya, hasil ekpansi, yaitu $E(R_{i-1})$, yang panjangnya 48 bit di-XOR-kan dengan K_i yang panjangnya 48 bit menghasilkan vektor A yang panjangnya 48-bit:

$E(R_{i-1}) \oplus K_i = A$

Vektor A dikelompokkan menjadi 8 kelompok, masing-masing 6 bit, dan menjadi masukan bagi proses substitusi. Proses substitusi dilakukan dengan menggunakan delapan buah kotak-S (*S-box*), S_1 sampai S_8. Setiap kotak-S menerima masukan 6 bit dan menghasilkan keluaran 4 bit. Kelompok 6-bit pertama menggunakan S_1, kelompok 6-bit kedua menggunakan S_2, dan seterusnya. Kedelapan kotak-S tersebut adalah:

S_1:

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1	1	8	2	4	9	1	7	5	1	3	1	1	0	6	1

5	2							1		4	0			3	

S_2:

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3:

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4:

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5:

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	16
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6:

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6

4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7:

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8:

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Keluaran proses substitusi adalah vektor *B* yang panjangnya 48 bit. Vektor *B* menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S. Permutasi dilakukan dengan menggunakan matriks permutasi *P* (*P-box*) sebagai berikut:

16	7	20	21	29	12	28	17	1	15	23	26	5	8	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Bit-bit $P(B)$ merupakan keluaran dari fungsi *f*. Akhirnya, bit-bit $P(B)$ di-XOR-kan dengan L_{i-1} untuk mendapatkan R_i (lihat Gambar 3.6):

$$R_i = L_{i-1} \oplus P(B)$$

Jadi, keluaran dari putaran ke-*i* adalah

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus P(B))$$

3.3.4 Permutasi Terakhir (*Inverse Initial Permutation*)

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Proses permutasi menggunakan matriks permutasi awal balikan (*inverse initial permutation* atau IP^{-1}) sebagai berikut:

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

3.3.5 Deskripsi

Proses deskripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan deskripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K_1, K_2, …, K_{16}, maka pada proses deskripsi urutan kunci yang digunakan adalah K_{16}, K_{15}, …, K_1.

Untuk tiap putaran 16, 15, …, 1, keluaran pada setiap putaran *deciphering* adalah

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

yang dalam hal ini, (R_{16}, L_{16}) adalah blok masukan awal untuk deciphering. Blok (R_{16}, L_{16}) diperoleh dengan mempermutasikan cipherteks dengan matriks permutasi IP^{-1}. Pra-keluaran dari *deciphering* adalah adalah (L_0, R_0). Dengan permutasi awal IP akan didapatkan kembali blok plainteks semula.

Tinjau kembali proses pembangkitan kunci internal pada Gambar 4. Selama *deciphering*, K_{16} dihasilkan dari (C_{16}, D_{16}) dengan permutasi PC-2. Tentu saja (C_{16}, D_{16}) tidak dapat diperoleh langsung pada permulaan *deciphering*. Tetapi karena (C_{16}, D_{16}) = (C_0, D_0), maka K_{16} dapat dihasilkan dari (C_0, D_0) tanpa perlu lagi melakukan pergeseran bit. Catatlah bahwa (C_0, D_0) yang merupakan bit-bit dari kunci eksternal *K* yang diberikan pengguna pada waktu deskripsi.

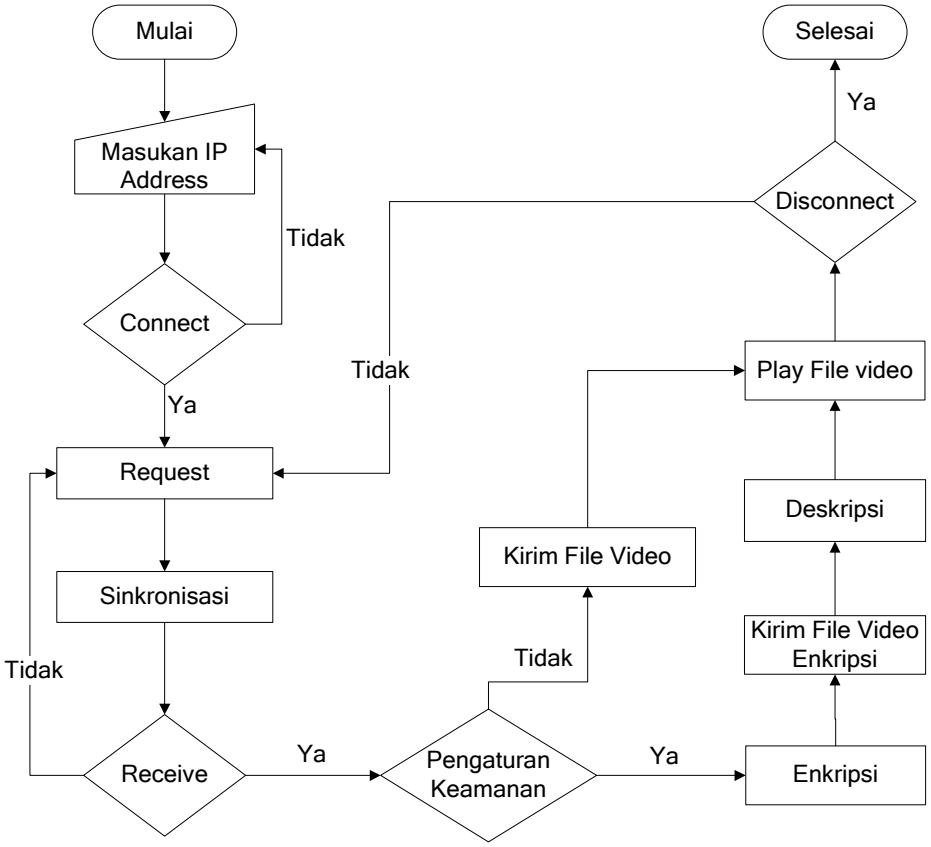
Selanjutnya, K_{15} dihasilkan dari (C_{15}, D_{15}) yang mana (C_{15}, D_{15}) diperoleh dengan menggeser C_{16} (yang sama dengan C_0) dan D_{16} (yang sama dengan C_0) satu bit ke kanan. Sisanya, K_{14} sampai K_1 dihasilkan dari (C_{14}, D_{14}) sampai (C_1, D_1). Catatlah bahwa (C_{i-1}, D_{i-1}) diperoleh dengan menggeser C_i dan D_i dengan cara yang sama seperti pada Tabel 1, tetapi pergeseran kiri (*left shift*) diganti menjadi pergeseran kanan (*right shift*).

3.4 Perancangan sistem

Perancangan sistem merupakan tindak lanjut dari tahap analisa. Perancangan sistem bertujuan untuk memberikan gambaran sistem yang akan dibuat.

3.4.1 Flowcart perancangan program aplikasi

Dibawah ini adalah diagram flowcart sebagai gambaran program aplikasi yang dibangun untuk client. Berikut adalah gambar perancangan antarmuka server.



Gambar 3. 3 flowchart program enkripsi data multimedia

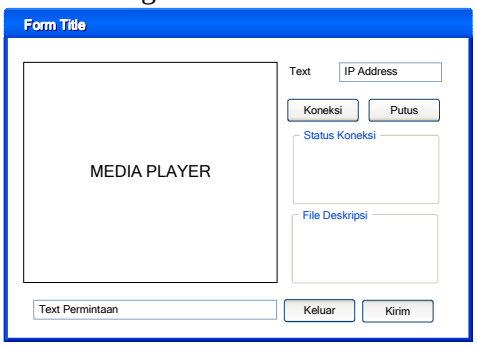
3.5 Perancangan Desain Antarmuka

Perancangan antarmuka dalam program aplikasi ini ada dua macam yaitu perancangan antarmuka untuk server dan perancangan antarmuka



Gambar 3. 4 Perancangan Antarmuka Server

Sedangkan perancangan antarmuka untuk aplikasi client adalah sebagai berikut:



Gambar 3. 5 Perancangan Antarmuka Client

BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap menterjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan implementasi aplikasi keamanan transmisi data multimedia menggunakan kryptografi *Data Encryption Standard* (DES).

4.2 Implementasi Kebutuhan Sistem

Dibutuhkan perangkat keras dan perangkat lunak yang dapat mendukung agar proses implementasi program yang dijalankan dapat dioperasikan demi mencapai hasil yang sesuai dengan rancangan.

4.2.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang dipasang pada sistem komputer yang digunakan untuk membangun dan mengimplementasikan aplikasi ini adalah sebagai berikut:

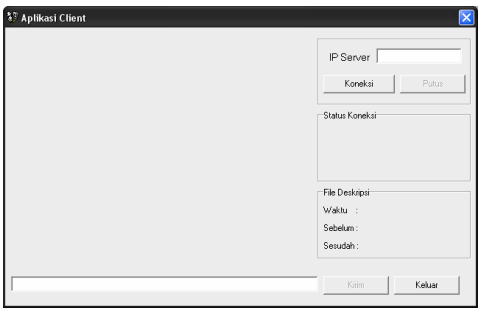
1. Sistem operasi Windows XP
2. Borlan Delphi 5
3. DU Meter
4. Wireshark 0.99.7

4.3 Implementasi Program

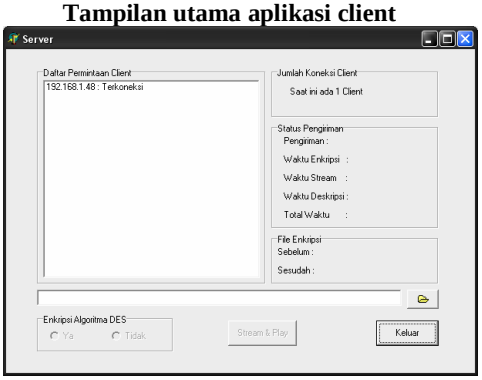
Implementasi program adalah mewujudkan hasil rancangan menjadi sebuah progam aplikasi yang dapat dioperasikan demi mencapai hasil yang sesuai dengan rancangan. Setelah melakukan tahap analisis dan perancangan, maka langkah selanjutnya adalah penerapan hasil perangkat yang akan diuji cobakan.

4.3.1 Tamplian Program

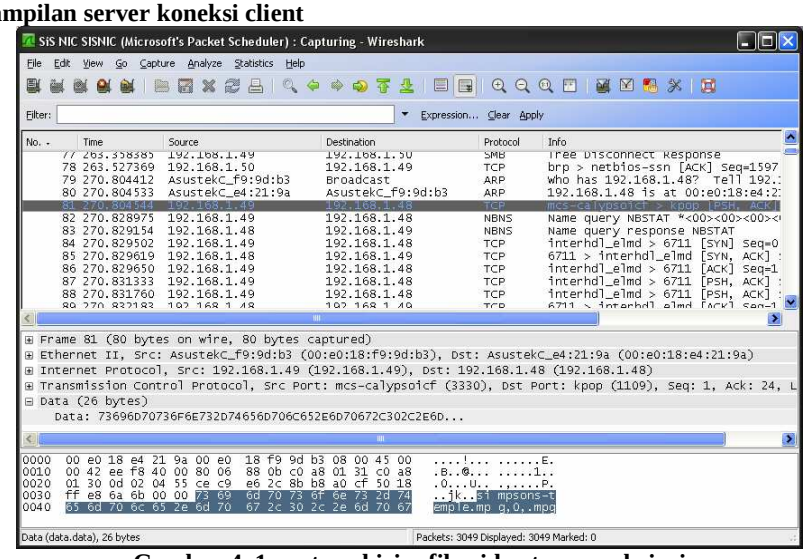
Tampilan utama aplikasi ini adalah seperti yang diperlihatkan



Tampilan utama aplikasi client



Tampilan server koneksi client



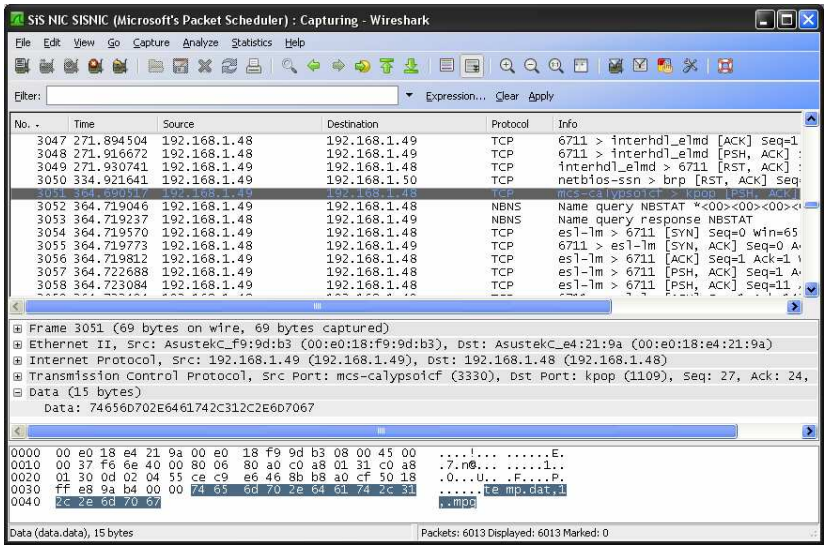
Gambar 4. 1 capture kirim file video tanpa enkripsi

Tampilan pada gambar 4.18 adalah hasil tangkapan pengiriman file video yang tidak dienkripsi, dari hasil tangkapan tersebut terlihat nama file yang dikirim yaitu ‘simpsons-temple.mpg’, angka 0 mengindikasikan bahwa file tersebut tidak terenkripsi dan ‘.mpg’ adalah ekstensi dari file yang dikirim tersebut.

4.4 Pengujian Sistem

4.4.1 Pengujian Keamanan Algoritma DES pada File Video

Pengujian keamanan dalam aplikasi ini sangat penting karena untuk mengukur sejauh mana.algoritma DES ini dalam mengamankan transmisi data khususnya dalam aplikasi file video, untuk menguji keamanan pengiriman dalam aplikasi ini diperlukan *software* tambahan yaitu Wireshark.



Gambar 4. 2 Capture kirim file video terenkripsi

Tampilan pada gambar 4.19 adalah hasil tangkapan pengiriman file video yang dienkripsi, dari hasil tangkapan tersebut bahwa nama file asli yang dikirim telah dirubah kedalam nama ‘temp.dat’, angka 1 mengindikasikan bahwa file tersebut terenkripsi, dan ‘.mpg’ adalah nama ekstensi dari file asli tersebut.

Panjang kunci eksternal DES hanya 64 bit atau 8 karakter, itupun yang dipakai hanya 56 bit. Tetapi, dengan panjang kunci 56 bit akan terdapat 2^{56} atau 72.057.594.037.927.936 kemungkinan kunci. Jika diasumsikan serangan *exhaustive key search* dengan menggunakan prosesor paralel mencoba setengah dari jumlah kemungkinan kunci itu, maka dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang benar.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari analisa dan pengujian yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma DES dinyatakan aman dalam mengamankan transmisi data multimedia karena tidak bisa tertangkap oleh tools wireshark dan juga memerlukan waktu yang sangat lama untuk menemukan kunci yang benar.
2. Setelah melalui proses enkripsi, terjadi kenaikan ukuran pada file video terhadap ukuran semula, tetapi didalam sistem ini tidak menjadi pembebanan dengan adanya kenaikan ukuran file video ini karena kenaikan ukuran terbesar setelah proses enkripsi hanya

delapan byte dari ukuran semula.

5.2 Saran

Setelah dilakukan analisis, pengujian dan kesimpulan, masih terdapat kekurangan-kekurangan yang dapat dijadikan saran untuk pengembangan selanjutnya, yang meliputi:

1. Dapat mengirim berbagai format file video.
2. Aplikasi ini tidak hanya berjalan pada jaringan LAN saja tetapi lebih disarankan dapat berjalan melalui jaringan internet.
3. Implementasi apliksi ini dapat dilakukan dengan menggunakan lebih dari satu client.

DAFTAR PUSTAKA

[1] Andi Team. 2003, *Memahami Model Enkripsi dan Security Data,* Andi, Yogyakarta.

[2] Budhi Irawan. 2005, *Jaringan Komputer*, Graha Ilmu, Yogyakarta.

[3] Roger S. Pressman. 2002, *Rekayasa Perangkat Lunak*, Andi, Yogyakarta.

[4] Martina, I. 2001. *36 Jam Belajar Komputer ; Delphi 5.0*, PT Elex Media Komputindo, Jakarta.

[5] William Stallings. 2001, Komunikasi Data dan Komputer, Salemba Teknika

[6] http://cairo.cs.uiuc.edu/ 10.30, 4 Oktober 2007, Raghavendra Vinayak Prabhu, pdf