BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini merupakan tahap penerjemahan kebutuhan pembangunan aplikasi ke dalam representasi perangkat lunak sesuai dengan hasil analisis yang telah dilakukan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan - kekurangan pada aplikasi yang baru untuk selanjutnya diadakan pengembangan sistem.

4.1 Implementasi

Tahap implementasi sistem merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin, serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Seluruh kode program yang digunakan dalam pembuatan Aplikasi Perbandingan Algoritma *Rijndael* dan Algoritma *Twofish* Pada Proses Pengiriman Data Teks Menggunakan Jaringan LAN (*Local Area Network*) yaitu menggunakan *Visual Basic* 6.0.

4.1.1 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada sistem komputer yang digunakan untuk membangun Aplikasi Perbandingan Algoritma *Rijndael* dan Algoritma *Twofish* Pada Proses Pengiriman Data Teks Menggunakan Jaringan *LAN* adalah sebagai berikut :

Table 4.1 Implementasi Perangkat Lunak

Perangkat Lunak	**Pengguna**
Sistem Operasi	*Windows XP Professional Service Pack* II
DBMS	*Ms. Access*
Visual Basic 6.0	✓
Uji *Security*	*Cain & Abel v*4.9.40

4.1.2 Implementasi Perangkat Keras

Kebutuhan minimum perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat adalah perangkat keras komputer PC *kompatibel* dengan spesifikasi yang disebutkan dibawah ini. Semakin tinggi spesifikasi komputer yang digunakan untuk menjalankan aplikasi, akan semakin baik. Kebutuhan minimumnya yaitu :

Table 4.2 Implementasi Perangkat Keras

Perangkat Keras	**Pengguna**	**Pembuat**
Processor	*Processor Pentium 4*	*Processor Core i3 Ghz 3.5 Ghz*
Harddisk	40 *GB*	80 *GB*
Ram	256 *MB*	1024 *MB*
VGA	32 *MB*	128 *MB*
Monitor	✓	✓
Mouse	✓	✓
Keyboard	✓	✓
Kabel *UTP*	✓	✓

4.1.3 Implementasi Basisdata (*Database*)

Dalam pembuatan aplikasi perbandingan algoritma *rijndael* dan algoritma *twofish* pada proses pengiriman data teks menggunakan jaringan *LAN* ini menggunakan *Ms.Access* sebagai *DBMS*.

Table 4.3 Nama Tabel Beserta *File* Implementasinya

No	Nama Tabel	Nama *file*
1	*Login*	kripto.mdb

Langkah pertama dalam pembuatan aplikasi ini yaitu pembuatan *database*nya, berikut adalah tabel yang dibentuk :

1. Tabel *Login*

   Tabel ini berfungsi untuk menampung data *user*.

   CREATE TABLE T*Login*
   `username` Text(50) *NOT NULL*,
   `pass` Text(16) *NOT NULL*,
   PRIMARY KEY (*`username`*);

4.1.4 Implementasi Antarmuka

Setelah melakukan implementasi data dilakukanlah implementasi antarmuka yang berguna sebagai tampilan untuk pengguna sistem. Implementasi dilakukan pada setiap hasil perancangan antarmuka yang telah dibuat kedalam bentuk aplikasi pada *Visual Basic* 6.0.

Untuk mengetahui lebih jelas tentang proses yang ada didalam sistem beserta hasil implementasinya dapat dilihat dalam tabel dibawah ini.

Table 4.4 Implementasi Antarmuka Pengirim

Menu	Deskripsi	Nama File
Kirim File	Digunakan untuk melakukan proses enkripsi dan kirim file	*Encrypt.frm*
Send IP	Digunakan untuk melakukan pengiriman IP pengirim	*SendIP.frm*
About	Melihat profil programmer	*About.frm*

Table 4.5 Implementasi Antarmuka Penerima

Menu	Deskripsi	Nama File
Terima File	Digunakan untuk menerima dan melakukan proses dekripsi file	*Decrypt.frm*
Receive IP	Digunakan untuk melakukan pengecekan dan penerimaan IP pengirim	*ReceiveIP.frm*
Login	Untuk masuk ke dalam menu Terima File	*FormLogin.frm*
About	Lihat profil programmer	*About.frm*

4.2 Pengujian Alpha

Sebuah aplikasi yang akan diterapkan haruslah bebas dari kesalahan-kesalahan atau *error*. Oleh karena itu, aplikasi harus diuji coba terlebih dahulu untuk menemukan kesalahan-kesalahan yang mungkin terjadi.

4.2.1 Metode Pengujian

Pengujian program ini menggunakan metode *Black Box*. Pengujian *Black Box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *Black Box* dilakukan dengan cara memberikan sejumlah *input* pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi menghasilkan *output* yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari *input* yang diberikan proses menghasilkan *output* yang sesuai dengan kebutuhan fungsionalnya, maka aplikasi yang bersangkutan telah benar, tetapi jika

output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka akan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

4.2.2 Rencana Pengujian

Rencana pengujian aplikasi perbandingan algoritma *rijndael* dan algoritma *twofish* pada proses pengiriman data teks menggunakan jaringan *LAN* terdiri dari pengirim dan penerima, rancangan semuanya menggunakan rencana pengujian dengan metode *Black Box* selengkapnya dapat dilihat pada tabel di bawah ini.

Rencana Pengujian Aplikasi Perbandingan Algoritma *Rijndael* dan Algoritma *Twofish* Pada Proses Pengiriman Data Teks Menggunakan Jaringan *LAN*.

Table 4.6 Rencana Pengujian Aplikasi Perbandingan Algoritma *Rijndael* dan *Twofish* Pada Proses Pengiriman Data Teks Menggunakan Jaringan *LAN*

Pengujian Aplikasi Perbandingan Algoritma *Rijndael* dan Algoritma *Twofish* Pada Proses Pengiriman Data Teks Menggunakan Jaringan *LAN* (*Local Area Network*)		
Item Uji	**Detail Uji**	**Jenis Uji**
Login	Verifikasi *Username*	*Black box*
	Verifikasi *Password*	*Black box*
Kirim *File*	Isi *File* Kirim	*Black box*
	Pilih *RadioButton Rijndael*	*Black box*
	Pilih *RadioButton Twofish*	*Black*

		box
	Isi *IP* Tujuan	*Black box*
	Sambung Koneksi	*Black box*
	Kirim *File*	*Black box*
	Indikator *List* Koneksi	*Black box*
	Indikator *Progress* Enkripsi	*Black box*
	Indikator *Progress* Kirim	*Black box*
	Indikator Waktu Pengiriman	*Black box*
	Indikator Ukuran *File*	*Black box*
	Indikator Waktu Enkripsi *Rijndael*	*Black box*
	Indikator Waktu Enkripsi *Twofish*	*Black box*
	Indikator *Icon File*	*Black box*
	Indikator *IP*	*Black box*
Terima *File*	Terima Koneksi	*Black box*
	Indikator *List* Koneksi	*Black box*
	Isi Terima *File*	*Black box*
	Pilih *RadioButton Twofish*	*Black box*
	Pilih *RadioButton Rijndae*l	*Black box*
	Dekripsi	*Black box*
	Indikator *Progress Twofish*	*Black box*
	Indikator *Progress Rijndael*	*Black box*
	Indikator Ukuran *File*	*Black box*
	Indikator Waktu Dekripsi *Rijndael*	*Black box*

	Indikator Waktu Dekripsi *Twofish*	*Black box*
	Indikator *Icon File*	*Black box*
	Indikator *IP*	*Black box*
Send IP	Mulai Koneksi	*Black box*
	Berhenti Koneksi	*Black box*
	Indikator *IP*	*Black box*
	Indikator *Status Bar*	*Black box*
Receive IP	Start IP	*Black box*
	Stop IP	*Black box*
	Indikator IP	*Black box*
	Indikator Status Bar	*Black box*

4.2.3 Kasus dan Hasil Pengujian

4.2.3.1 Pengujian Pengirim

1. Pengujiuan Kirim *File*

Tabel 4.7 Pengujian Kirim *File*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Isi *File* Kirim	Tampil *CommonDialog* yang mempunyai ekstensi *file* berupa *.doc, .pdf, .txt*	Dapat menampilkan *CommonDialog* yang mempunyai ekstensi *file* berupa *.doc, .pdf, .txt*	[√] Diterima [ ] Ditolak
Pilih *RadioButton Rijndael*	Jika *file* sudah terisi atau benar dalam pengisian Pilih	File sudah terisi atau benar dalam pengisian Pilih	[√] Diterima [ ] Ditolak

	File, maka akan dapat langsung memilih algoritma *Rijndael*	*File*, maka akan dapat langsung memilih algoritma *Rijndael*	
Pilih *RadioButton Twofish*	Jika *file* sudah terisi atau benar dalam pengisian Pilih *File*, maka akan dapat langsung memilih algoritma *Twofish*	File sudah terisi atau benar dalam pengisian Pilih *File*, maka akan dapat langsung memilih algoritma *Twofish*	[√] Diterima [ ] Ditolak
Isi *IP* Tujuan	Dapat terkoneksi dengan benar jika *IP* Tujuan sudah terisi dengan benar	*IP* Tujuan sudah dapat terkoneksi dengan benar	[√] Diterima [ ] Ditolak
Sambung Koneksi	Data tidak dapat dikirim sebelum tombol proses Sambung Koneksi di klik	Data sudah siap dikirim setelah tombol proses Sambung Koneksi di klik	[√] Diterima [ ] Ditolak
Enkripsi	Jika persyaratan dalam mengirim *file* sudah terpenuhi maka boleh langsung klik enkripsi	Sudah bisa langsung enkripsi ketika persyaratan sudah terpenuhi	[√] Diterima [ ] Ditolak
Indikator *List* Koneksi	*List* Koneksi akan berfungsi jika dalam menginputkan *IP* Tujuan sudah benar	*Input IP* Tujuan sudah benar dan info *List* Koneksi terpenuhi	[√] Diterima [ ] Ditolak
Indikator *Progress* Enkripsi	Indikator *Progress* Enkripsi akan berjalan jika sudah sesuai dengan pilihan pada *RadioButton Rijndael* atau *Twofish*	Indikator *progress* Enkripsi sudah bisa berjalan setelah dipilih salah satu *RadioButton Rijndael* atau *Twofish*	[√] Diterima [ ] Ditolak
Indikator Progress Kirim	Indikator *Progress* Kirim akan berjalan jika *progress*	Indikator *progress* Kirim sudah bisa	[√] Diterima [ ] Ditolak

	enkripsi sudah selesai	berjalan setelah *progress* enkripsi selesai	
Indikator Waktu Pengiriman	Indikator Waktu Pengiriman akan muncul jika *progress* kirim sudah selesai	Indikator Waktu Pengiriman muncul saat *progress* Kirim selesai	[√] Diterima [ ] Ditolak
Indikator Ukuran *File*	Indikator Ukuran *File* akan muncul jika sudah memilih *file* yang akan di enkripsi dan dikirim	Indikator Ukuran *File* muncul saat *file* yang akan di enkripsi dan dikirim sudah dipilih	[√] Diterima [ ] Ditolak
Indikator Waktu Enkripsi *Rijndael*	Indikator Waktu Enkripsi *Rijndael* akan muncul pada saat *progress* enkripsi selesai	Indikator Waktu Enkripsi *Rijndael* muncul saat *progress* enkripsi selesai	[√] Diterima [ ] Ditolak
Indikator Waktu Enkripsi *Twiofish*	Indikator Waktu Enkripsi *Twofish* akan muncul pada saat *progress* enkripsi selesai	Indikator Waktu Enkripsi *Twofish* muncul saat *progress* enkripsi selesai	[√] Diterima [ ] Ditolak
Indikator *Icon File*	Indikator *Icon File* akan muncul pada saat pemilihan *file* dengan ekstensi *.doc, .pdf, .txt*	Indikator *Icon File* muncul saat pemilihan *file* dengan ekstensi *.doc,.pdf,.txt* selesai	[√] Diterima [ ] Ditolak
Indikator *IP*	Indikator *IP* akan muncul pada saat muncul *form* Kirim *File*	Indikator *IP* muncul setelah *form* Kirim *File* di eksekusi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih File Kosong	Data tidak dapat di proses dan menampilkan pesan "Pilih File Yang Akan Dienkripsi dan Dikirim"	Pengisian data gagal dan menampilkan pesan "Pilih File Yang Akan Dienkripsi dan Dikirim"	[√] Diterima [ ] Ditolak
RadioButton	Data tidak dapat	Enkripsi data	[√] Diterima

Rijndael dan *Twofish* Kosong	dienkripsi dan menampilkan pesan "Pilih Salah Satu Algoritma yang Akan Dienkripsi"	gagal dan menampilkan pesan "Pilih Salah Satu Algoritma yang Akan Dienkripsi"	[ ] Ditolak
IP Tujuan Kosong	Data tidak dapat dikirim dan menampilkan pesan "Masukkan *IP* Tujuan"	Data gagal dikirim dan menampilkan pesan "Masukkan *IP* Tujuan"	[√] Diterima [ ] Ditolak

2. Pengujian *Send IP*

Tabel 4.8 Pengujian *Send IP*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mulai Koneksi	Tampil *textbox* yang mampu menampilkan *IP* dalam keadaan mulai	Dapat menampilkan *textbox* yang mampu menampilkan *IP*	[√] Diterima [ ] Ditolak
Berhenti Koneksi	Tampil *textbox* yang mampu menampilkan *IP* dalam keadaan berhenti	Dapat menampilkan *textbox* yang mampu menampilkan *IP*	[√] Diterima [ ] Ditolak
Indikator *IP*	Indikator *IP* akan muncul pada saat muncul *form Send IP*	Indikator *IP* muncul setelah *form Send IP* di eksekusi	[√] Diterima [ ] Ditolak
Indikator Status Bar	Indikator Status Bar muncul pada saat tombol mulai koneksi atau berhenti koneksi dijalankan	Indikator Status Bar muncul setelah tombol mulai koneksi atau berhenti koneksi dijalankan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
IP Address Kosong	*Form Send IP* tidak	Koneksi terputus	[√] Diterima

	akan berfungsi dan tersambung	sehingga tidak akan berfungsi atau tersambung	[ ] Ditolak

4.2.3.2 Pengujian Penerima

1. Pengujian *Login*

Tabel 4.9 Pengujian *Login*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : ridhky *Password* : admin	Dapat terisi pada *textfield username* dan *textfield password*	Dapat mengisi nama *usename* dan *password* Sesuai yang diharapakan	[√] Diterima [ ] Ditolak
Login	Data admin dicari di tabel *login*	Tombol *login* dapat berfungsi Sesuai dengan yang diharapkan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *username* dan *password* tidak terdaftar, Data username kosong, Data *password* kosong, dan data kedua-duanya kosong	Tidak dapat *login* dan menampilkan pesan "Masukkan *Username* dan/ atau Masukkan *Password* "	*User* tidak dapat *login* dan mendapatkan pesan "Masukkan *Username* dan/ atau Masukkan *Password*" Sesuai yang diharapkan	[√] Diterima [ ] Ditolak

2. Pengujian Terima *File*

Tabel 4.10 Pengujian Terima *File*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Terima Koneksi	Data tidak dapat diterima sebelum	Data sudah siap diterima setelah	[√] Diterima

	tombol proses Terima Koneksi di klik	tombol proses Terima Koneksi di klik	[ ] Ditolak
Indikator *List* Koneksi	*List* Koneksi akan berfungsi jika sudah klik tombol Terima Koneksi	*List* Koneksi berfungsi setelah mengklik tombol Mulai Koneksi	[√] Diterima [ ] Ditolak
Pilih *RadioButton Twofish*	Jika *file* sudah terisi atau benar dalam pengisian Terima *File*, maka akan dapat langsung memilih algoritma *Twofish*	*File* sudah terisi atau benar dalam pengisian Terima *File*, maka akan dapat langsung memilih algoritma *Twofish*	[√] Diterima [ ] Ditolak
Pilih *RadioButton Rijndael*	Jika *file* sudah terisi atau benar dalam pengisian Terima *File*, maka akan dapat langsung memilih algoritma *Rijndael*	*File* sudah terisi atau benar dalam pengisian Terima *File*, maka akan dapat langsung memilih algoritma *Rijndael*	[√] Diterima [ ] Ditolak
Dekripsi	Jika persyaratan dalam mengirim *file* sudah terpenuhi maka boleh langsung klik Dekripsi	Sudah bisa langsung Dekripsi ketika persyaratan sudah terpenuhi	[√] Diterima [ ] Ditolak
Indikator *Progress Twofish*	Indikator *Progress Twofish* akan berjalan jika sudah sesuai dengan pilihan pada *RadioButton Twofish*	Indikator *progress Twofish* sudah bisa berjalan setelah dipilih salah satu *RadioButton Twofish*	[√] Diterima [ ] Ditolak
Indikator *Progress Rijndael*	Indikator *Progress Rijndael* akan berjalan jika sudah sesuai dengan pilihan pada *RadioButton Rijndael*	Indikator *progress Rijndael* sudah bisa berjalan setelah dipilih salah satu *RadioButton Rijndael*	[√] Diterima [ ] Ditolak
Indikator Ukuran *File*	Indikator Ukuran *File* akan muncul jika sudah memilih	Indikator Ukuran *File* muncul saat *file* yang akan di	[√] Diterima [ ] Ditolak

	file yang akan di dekripsi	dekripsi dan sudah dipilih	
Indikator Waktu Dekripsi *Rijndael*	Indikator Waktu Dekripsi *Rijndael* akan muncul pada saat *progress* dekripsi selesai	Indikator Waktu Dekripsi *Rijndael* muncul saat *progress* dekripsi selesai	[√] Diterima [ ] Ditolak
Indikator Waktu Dekripsi *Twofish*	Indikator Waktu Dekripsi *Twofish* akan muncul pada saat *progress* dekripsi selesai	Indikator Waktu Dekripsi *Twofish* muncul saat *progress* dekripsi selesai	[√] Diterima [ ] Ditolak
Indikator *Icon File*	Indikator *Icon File* akan muncul pada saat pemilihan *file* dengan ekstensi *.doc, .pdf, .txt*	Indikator *Icon File* muncul saat pemilihan *file* dengan ekstensi *.doc,.pdf,.txt* selesai	[√] Diterima [ ] Ditolak
Indikator *IP*	Indikator *IP* akan muncul pada saat muncul *form* Terima *File*	Indikator *IP* muncul setelah *form* Terima *File* di eksekusi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Hasil *File* Kosong	Data tidak dapat di proses dan menampilkan pesan "*File* Yang Akan Di*Decrypt* Tidak Ada"	Pengisian data gagal dan menampilkan pesan "*File* Yang Akan Di*Decrypt* Tidak Ada"	[√] Diterima [ ] Ditolak
RadioButton Rijndael dan *Twofish* Kosong	Data tidak dapat didekripsi dan menampilkan pesan "Pilih Salah Satu Algoritma yang Akan Didekripsi"	Dekripsi data gagal dan menampilkan pesan "Pilih Salah Satu Algoritma yang Akan Didekripsi"	[√] Diterima [ ] Ditolak

3. Pengujian *Receive IP*

Tabel 4.11 Pengujian *Receive IP*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Start IP	Tampil *textbox* yang mampu menampilkan *IP* dalam keadaan *start*	Dapat menampilkan *textbox* yang mampu menampilkan *IP*	[√] Diterima [ ] Ditolak
Stop IP	Tampil *textbox* yang mampu menampilkan *IP* dalam keadaan *stop*	Dapat menampilkan *textbox* yang mampu menampilkan *IP*	[√] Diterima [ ] Ditolak
Indikator *IP*	Indikator *IP* akan muncul pada saat muncul form *Receive IP*	Indikator *IP* muncul setelah form *Receive IP* di eksekusi	[√] Diterima [ ] Ditolak
Indikator Status Bar	Indikator Status Bar muncul pada saat tombol *start IP* atau *Stop IP* dijalankan	Indikator Status Bar muncul setelah tombol *Start IP* atau *Stop IP* dijalankan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
IP Address Kosong	*Form Receive IP* tidak akan berfungsi dan tersambung	Koneksi terputus sehingga tidak akan berfungsi atau tersambung	[√] Diterima [ ] Ditolak

4.2.4 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian sistem yang telah dilakukan secara keseluruhan memberikan kesimpulan bahwa proses pada aplikasi perbandingan algoritma rijndael dan algoritma twofish pada proses pengiriman data teks menggunakan jaringan LAN telah melalui tahap perbaikan dan sudah

dimaksimalkan terhadap proses-proses tersebut dan secara fungsional sistem sudah dapat digunakan dan menghasilkan *output* yang diharapkan.

4.3 Pengujian Data / Sistem

Pengujian data / sistem ini dilakukan pada perangkat lunak yang dilakukan secara bertahap. Adapun tujuan pengujiannya adalah sebagai berikut :

1. Menguji pengaruh sistem enkripsi dan dekripsi terhadap ukuran pada *file* teks yang berupa *.doc, .txt,* dan *.pdf* pada kinerja perangkat lunak.
2. Menguji performa dari proses enkripsi dan dekripsi yang ada pada *file* berupa *.doc, .txt,* dan *.pdf* pada kinerja perangkat lunak. Performa diukur dari penggunaan waktu.
3. Menguji apakah proses penyadapan akan menghasilkan teks acak (*ciphertext*) apabila mode enkripsi diaktifkan.

Pengujian dilakukan pada lingkungan yang sama dengan implementasi perangkat lunak.

4.3.1 Kasus Uji

Pengujian terbagi menjadi tiga tahap, yaitu pengujian ukuran, waktu, dan keamanan. Kasus uji pada pengujian ukuran dilakukan dengan menganalisis ukuran *file* teks sebelum dan sesudah proses enkripsi dan dekripsi ketika perangkat lunak dijalankan. Kasus uji yang kedua pada pengujian kecepatan dilakukan dengan menganalisis waktu setelah mengalami proses enkripsi dan dekripsi.

Sementara itu pengujian enkripsi dan dekripsi secara umum dilakukan dengan dua cara, yaitu pengujian pengiriman dan penerimaan teks. Pengujian

yang terakhir ditujukan untuk memastikan bahwa sistem enkripsi pada perangkat lunak dapat digunakan untuk mempersulit proses penyadapan. Untuk itu pengujian tersebut dilakukan dengan mencoba menyadap pesan yang dikirim melalui suatu perangkat lunak yang dapat menangkap paket-paket data teks tersebut.

4.3.2 Pengujian Ukuran

Pengujian ukuran dilakukan dengan membandingkan kapasitas dari beberapa *file* teks berupa *.doc, .txt*, dan *.pdf* pada awal sebelum dan sesudah proses enkripsi dan dekripsi dengan menggunakan algoritma *Rijndael* dan *Twofish* ketika perangkat lunak dijalankan. Berikut ini adalah Tabel 4.12 yang berisi hasil pengujian yang dilakukan pada sebuah komputer 1 dengan spesifikasi *AMD Athlon™ 64 Processor 3500+, 2.19 GHz* dengan *RAM 1.50 GB.*

Tabel 4.12 Hasil Pengujian Berdasarkan Ukuran *File* Pada Komputer 1

No	Nama *File*	Ukuran *File* Asli (*bytes*)	Panjang Kunci	Ukuran Setelah Enkripsi (*bytes*)		Ukuran Setelah Dekripsi (*bytes*)	
				Rijndael	*Twofish*	*Rijndael*	*Twofish*
1.	Laporan spesifikasi. doc	23,701	16	23,720	23,728	23,701	23,701
2.	Abstrak.txt	3,021	16	3,040	3,040	3,021	3,021
3.	GL02*Samp le*.pdf	383,318	16	383,336	383,344	383,318	383,318

Berikut ini adalah Tabel 4.13 yang berisi hasil pengujian yang dilakukan pada sebuah komputer 2 dengan spesifikasi *Intel ® Pentium ® Dual CPU E2140, 1.60 GHz* dengan *RAM 2.00 GB.*

Tabel 4.13 Hasil Pengujian Berdasarkan Ukuran *File* Pada Komputer 2

No	Nama *File*	Ukuran *File* Asli (*bytes*)	Panjang Kunci	Ukuran Setelah Enkripsi (*bytes*)		Ukuran Setelah Dekripsi (*bytes*)	
				Rijndael	*Twofish*	*Rijndael*	*Twofish*
1.	Laporan spesifikasi.doc	23,701	16	23,720	23,728	23,701	23,701
2.	Abstrak.txt	3,021	16	3,040	3,040	3,021	3,021
3.	GL02*Sample*.pdf	383,318	16	383,336	383,344	383,318	383,318

Berdasarkan hasil pengujian diatas dapat dilihat bahwa *file* dengan ukuran kecil maupun besar tidak beda jauh selisihnya setelah melalui proses enkripsi dengan algoritma *Rijndael* dan *Twofish* pada spesifikasi *hardware* komputer yang berbeda pula. Sedangkan untuk proses dekripsi, *file* dengan ukuran kecil maupun besar akan tetap sama seperti *file* aslinya.

4.3.3 Pengujian Waktu

Pengujian ini dilakukan untuk mengukur kecepatan proses enkripsi dan dekripsi dari kedua algoritma tersebut. *File - file* dipilih dari jenis yang berbeda-beda yaitu berupa *.doc, .txt,* dan *.pdf*. Hal terpenting dalam pengujian ini adalah terdapat perbedaan ukuran *file* dan spesifikasi *hardware* komputer. Sehingga dapat dilihat kecepatan proses enkripsi dan dekripsi dengan data yang berbeda-beda ukuran dan spesifikasi *hardware* komputernya. Berikut ini adalah Tabel 4.14 yang berisi hasil pengujian yang dilakukan pada sebuah komputer 1 dengan spesifikasi *AMD Athlon™ 64 Processor 3500+, 2.19 GHz* dengan *RAM 1.50 GB*.

Tabel 4.14 Hasil Pengujian Berdasarkan Waktu Enkripsi dan Dekripsi Pada Komputer 1

No	Nama *File*	Ukuran *File* Asli (*bytes*)	Panjang Kunci	Waktu Enkripsi (hh:mm:ss)		Waktu Dekripsi (hh:mm:ss)	
				Rijndael	*Twofish*	*Rijndael*	*Twofish*
1.	Laporan spesifikasi.doc	23,701	16	00:00:06	00:00:04	00:00:07	00:00:04
2.	Abstrak.txt	3,021	16	00:00:01	00:00:01	00:00:01	00:00:01
3.	GL02*Sample*.pdf	383,318	16	00:01:08	00:00:56	00:01:09	00:00:57

Berikut ini adalah Tabel 4.15 yang berisi hasil pengujian yang dilakukan pada sebuah komputer 2 dengan spesifikasi *Intel ® Pentium ® Dual CPU E2140, 1.60 GHz* dengan *RAM 2.00 GB.*

Tabel 4.15 Hasil Pengujian Berdasarkan Waktu Enkripsi dan Dekripsi Pada Komputer 2

No	Nama *File*	Ukuran *File* Asli (*bytes*)	Panjang Kunci	Waktu Enkripsi (hh:mm:ss)		Waktu Dekripsi (hh:mm:ss)	
				Rijndael	*Twofish*	*Rijndael*	*Twofish*
1.	Laporan spesifikasi.doc	23,701	16	00:00:01	00:00:01	00:00:01	00:00:01
2.	Abstrak.txt	3,021	16	00:00:01	00:00:01	00:00:01	00:00:01
3.	GL02*Sample*.pdf	383,318	16	00:00:08	00:00:08	00:00:08	00:00:08

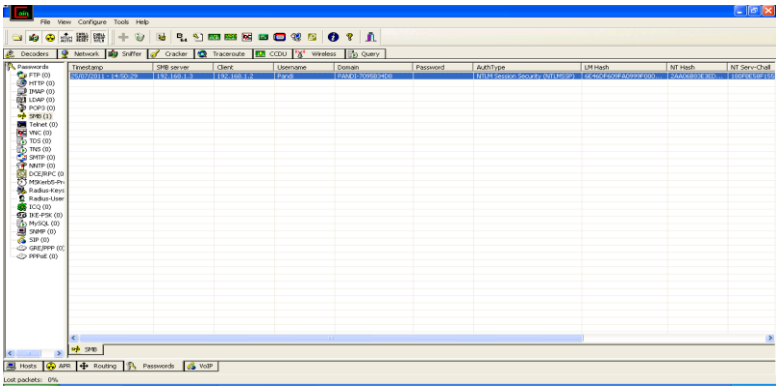
Berdasarkan hasil pengujian diatas dapat dilihat bahwa untuk waktu yang diperlukan dalam proses enkripsi dan dekripsi dengan algoritma *Rijndael* dan *Twofish* kurang lebih tidak jauh berbeda. Tetapi dengan adanya perbedaan dari spesifikasi *hardware* komputer 1 dan 2 itu sangat jauh berbeda selisih waktunya, karena memerlukan memori yang cukup besar dalam melakukan proses enkripsi dan dekripsi tersebut. Walaupun begitu *file* dengan ukuran yang lebih besar akan diproses lebih lama dibandingkan dengan *file* yang berukuran lebih kecil.

Berdasarkan keseluruhan data hasil eksperimen dapat ditarik kesimpulan bahwa algoritma *Rijndael* selalu berjalan lebih lambat daripada algoritma *Twofish*, karena algoritma *Rijndael* memiliki cara penjadwalan kunci dan pembangkitan *S-Box* yang cukup rumit dibandingkan dengan algoritma *Twofish*.

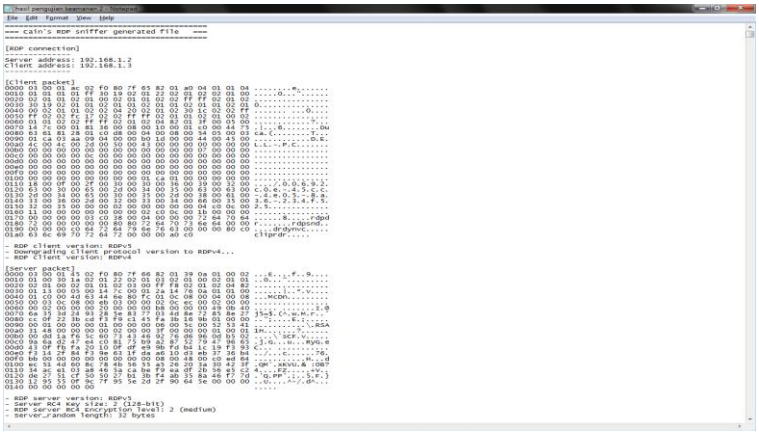
4.3.4 Pengujian Keamanan

Pengujian penyadapan ini dilakukan dengan menggunakan sebuah perangkat lunak yang bernama *Cain & Abel v4.9.40*. Perangkat lunak ini dikembangkan oleh *Cain & Abel* untuk menyadap segala sesuatu aktifitas yang terjadi dalam satu jaringan *local*. Penyadapan dapat dilakukan selama *client* yang disadap masih berada dalam satu jaringan *LAN* dengan penyadap.

Pada pengujian ini, pesan dicoba dikirimkan oleh perangkat lunak yang telah dibuat ketika aplikasi *Cain & Abel* sedang berjalan. Pada pengujian ini pesan yang akan dikirimkan dalam kondisi terenkripsi. Ketika pesan tersebut ditransmisikan ke sisi penerima, perangkat lunak penyadap dapat menangkap sebuah paket data pesan yang telah dikirim, namun isi teks pesan tersebut adalah teks pesan yang telah dienkripsi. *Screenshot* pengujian tersebut dapat dilihat pada gambar dibawah ini.



Gambar 4.1 Penyadapan Dengan Menggunakan *tools Cain & Abel*



Gambar 4.2 Penyadapan Pesan Terenkripsi

Seperti yang telah dilihat pada hasil pengujiannya, pengiriman pesan dengan mode enkripsi pada perangkat lunak yang telah dibuat, maka pesan yang ditangkap penyadap hanyalah berupa *ciphertext* yang perlu didekripsi terlebih dahulu untuk dibaca isinya. Dengan demikian dapat dikatakan bahwa mode enkripsi pada perangkat lunak ini akan mempersulit proses penyadapan.