PERANCANGAN DAN IMPLEMENTASI *SECURE CLOUD* DENGAN MENGGUNAKAN *DIFFIE-HELLMAN KEY EXCHANGE* DAN *TRIPLE DES ALGORITHM (3DES)*

DESIGN AND IMPLEMENTATION SECURE CLOUD BY USING DIFFIE-HELLMAN KEY EXCHANGE AND TRIPLE DES ALGORITHM (3DES)

Bernard Raditio Parulian[1], Surya Michrandi Nasution[2], Tito Waluyo Purboyo[3]

Telkom University
Bandung, Indonesia bernardraditio@students.telkomuniversity.ac.id[1],
michrandi@telkomuniversity.ac.id[2] titowaluyo@telkomuniversity.ac.id[3]

Abstrak

Perkembangan metode penyimpanan digital sekarang ini semakin beragam, salah satunya adalah metode penyimpanan berbasis *Cloud* yang memberikan akses kepada penggunannya untuk menyimpan data di dalam Internet dengan kapasitas penyimpanan yang dapat disesuaikan dengan keinginan penggunanya. Namun, metode *Cloud* ini memiliki kekurangan yaitu berkaitan dengan masalah keamanan data . Pada penelitian ini akan akan dibahas mengenai keamanan data pada *Cloud* dengan menggunakan kombinasi algoritma kriptografi *Triple DES Algorithm (3DES)* dan pertukaran kunci *Diffie-Hellman.* Sistem yang dibangun adalah aplikasi berbasis desktop yang menyediakan konten untuk mengunggah dan mengunduh *file* dokumen dari *user,* yang didalam nya sudah terdapat proses enkripsi dan dekripsi dengan algoritma kriptografi *Triple DES Algorithm (3DES)* serta pertukaran kunci user dengan *Diffie-Hellman.* Penelitian ini bertujuan untuk menganalisa performansi dari algoritma kriptografi *Triple DES Algorithm (3DES)* pada keamanan file dokumen saat ada proses enkripsi dan dekripsi, *avalanche effect,* pemakaian daya dan *Diffie-Hellman*.

Kata kunci: ***Cloud, Kriptografi, Diffie-Hellman Key Exchange Algorithm., Triple DES Algorithm (3DES), Secure Cloud***

Abstract

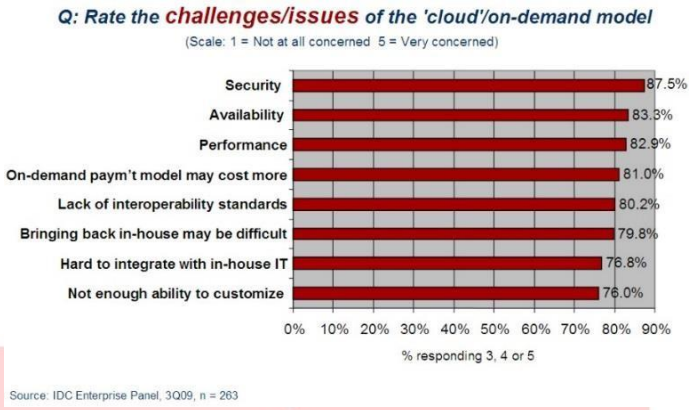
The development of digital storage methods are now more diverse, one of which is a cloud-based storage methods that provide access to the consumer to store the data in the Internet with a storage capacity that can be adapted to the wishes of its users. However, this method has the disadvantage that Cloud pertaining to data security issues. This research will be discussed regarding data security in cloud by using a combination of cryptographic algorithms Algorithm Triple DES (3DES) and Diffie-Hellman key exchange. The system is built is a desktop-based application that provides content to upload and download files from the user document, which was already contained in the encryption and decryption algorithm Triple DES cryptographic algorithm (3DES) as well as user key exchange with Diffie-Hellman. This study aims to analyze the performance of cryptographic algorithms Algorithm Triple DES (3DES) in the current document existing file security encryption and decryption process, avalanche effect, power consumption and Diffie-Hellman.

Keyword: ***Cloud, Cryptography, Diffie-Hellman Key Exchange Algorithm., Triple DES Algorithm (3DES), Secure Cloud***

1. Pendahuluan

Cloud computing merupakan salah satu solusi dalam penyimpanan data sekarang ini. *Cloud computing* adalah sebuah model yang memungkinkan penggunaan sumber daya (*resource*) secara bersama sama dan mudah, menyediakan jaringan akses di mana-mana, dapat dikonfigurasi, dan disesuaikan dengan kebutuhan[3].

Sejalan dengan berkembangnya teknologi komputasi saat ini, konsep *cloud computing* ini tidak lepas dari masalah – masalah yang berhubungan dengan keamanan data dari *file – file* yang disimpan di *server cloud* tersebut. Gambar dibawah ini menunjukkan ancaman pada *cloud service [5]*.



Gambar 1.1 Hasil Survei IDC terhadap ancaman pada *Cloud Service*.

Kriptografi menjadi salah satu solusi untuk mengatasi masalah keamanan di *cloud* . Kriptografi adalah ilmu untuk menjaga kerahasiaan informasi[4].. Salah satu algoritma yang dapat menjaga keamanan data adalah algoritma *Triple DES Algorithm(3DES). Triple DES Algorithm(3DES)* merupakan suatu algoritma pengembangan dari algoritma *DES* (*Data Encryption Standard*). *Triple DES Algorithm(3DES)* memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari *DES*)[2]. *3DES* mengenkripsi *plaintext* menjadi *ciphertext*. Saat ini *3DES* masih digunakan di beberapa aplikasi salah satunya adalah *payment system security*.

Untuk mengamankan pertukaran kunci salah satu algoritma kriptografi yang digunakan adalah algoritma *Diffie-Hellman Key Exchange*. *Diffie-Hellman Key Exchange* atau pertukaran kunci *Diffie-Hellman* adalah metode pertukaran kunci rahasia pada komunikasi menggunakan kriptografi simetris[8].

Pada penelitian tugas akhir ini, akan menggunakan kombinasi dari algoritma kriptografi *3DES* untuk enkripsi *file* pada *cloud system* dan algoritma *Diffie-Hellman Key Exchange* untuk pengamanan pertukaran kunci.

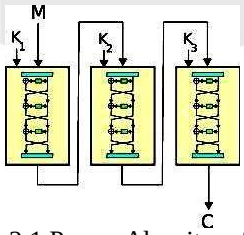
2. Tinjauan Pustaka

2.1 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan[4].Dalam kriptografi pesan asli disebut *plaintext* serta terdapat istilah enkripsi dan dekripsi. Enkripsi adalah proses yang melakukan perubahan sebuah kode dari yang dapat dimengerti menjadi sebuah kode yang tidak dapat dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *cipher*. Sedangkan, dekripsi adalah proses untuk mengembalikan informasi teracak menjadi bentuk aslinya dengan menggunakan algoritma yang sama pada saat mengenkripsi.

2.1.1 Algoritma Kriptografi *Triple Data Encryption Standard*

3DES (Triple Data Encryption Standard) merupakan suatu algoritma pengembangan dari algoritma *DES (Data Encryption Standard).* Pada dasarnya algoritma yang digunakan sama, hanya pada *3DES* dikembangkan dengan melakukan enkripsi dengan implementasi algoritma *DES* sebanyak tiga kali. *3DES* memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari *DES*). Pada algoritma *3DES* dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma *DES* [2].



Gambar 2.1 Proses Algoritma *3DES*[2]

Ada tiga pilihan untuk pemilihan kunci eksternal algoritma ini, yaitu:

a. K_1,K_2 dan K_3 adalah kunci berbeda
b. K_1 dan K_2 adalah kunci berbeda dan K_3=K_1
c. $K_1 = K_2 = K_3$

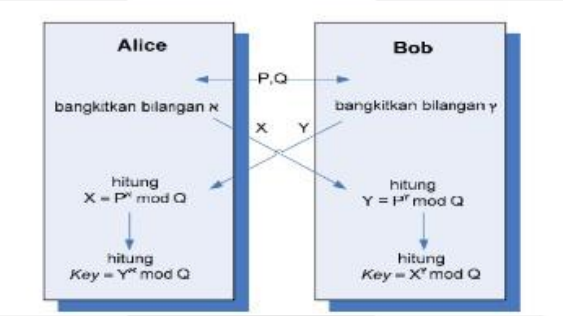
Proses enkripsi dan dekripsi algoritma *3DES* dapat memakai beberapa cara, yaitu:

Tabel 2.1 Cara Enkripsi dan Dekripsi 3DES [2]

Cara	Enkripsi	Dekripsi
1	DES –EDE2 • $K_1 \neq K_2$,$K_3=K_1$ • C=E[D{E(*Plaintext*,K_1),K_2},K_3]	DES –DED2 • $K_1 \neq K_2$,$K_3=K_1$ • P=D[E{D(*Ciphertext*,K_3),K_2},K_1]
2	DES – EEE2 • $K_1 \neq K_2$, $K_3 = K_1$ • C = E [E {E (*Plaintext*, K_1), K_2}, K_3]	DES –DDD2 • $K_1 \neq K_2$,$K_3=K_1$ • P=D[D{D(*Ciphertext*,K_3),K_2},K_1]
3	DES – EDE3 • $K_1 \neq K_2 \neq K_3 \neq K_1$ • C = E [D {E (*Plaintext*, K_1), K_2}, K_3]	DES –DED2 • $K_1 \neq K_2 \neq K_3 \neq K_1$ • P=D[E{D(*Ciphertext*,K_3),K_2},K_1]
4	DES – EEE3 • $K_1 \neq K_2 \neq K_3 \neq K_1$ • C = E [E {E (*Plaintext*, K_1), K_2}, K_3]	DES –DDD2 • $K_1 \neq K_2 \neq K_3 \neq K_1$ • P=D[E{D(*Ciphertext*,K_3),K_2},K_1]

2.1.2 Pertukaran Kunci Diffie-Hellman

Diffie-Hellman Key Exchange atau pertukaran kunci *Diffie-Hellman* adalah metode pertukaran kunci rahasia pada komunikasi menggunakan kriptografi simetris. Kekuatan dari metode ini adalah pada sulitnya melakukan perhitungan logaritma diskrit antara pengirim dan penerima kunci [8] .Disamping itu kekuatan Diffie-Hellman 3072 bit dan 7680 bit setara dengan keamanan AES 128 bit dan 192 bit [1]. Tahap-tahap pertukaran kunci *Diffie-Hellman* adalah sebagai berikut :



Gambar 2.2 Langkah-langkah di pertukaran kunci Diffie-Hellman[9].

1. Misalkan Alice dan Bob adalah pihak-pihak yang berkomunikasi. Mula-mula Alice dan Bob menyepakati 2 buah bilangan yang besar (sebaiknya prima) dan sedemikian sehingga < Nilai dan tidak perlu rahasia,
2. Alice membangkitkan bilangan bulat acak x yang besar dan mengirim hasil perhitungan berikut kepada Bob,
   =
3. Bob membangkitkan bilangan bulat acak y yang besar dan mengirim hasil perhitungan, berikut kepada Alice,
   =
4. Alice lalu mendapatkan kunci dari,
   =
5. Bob mendapatkan kunci dari,
   =

2.2 *Cloud Computing*

Menurut NIST (*National Institute of* Standard *and Technology*) mendefinisikan *cloud computing* sebagai sebuah model yang memungkinkan adanya penggunaan sumber daya (*resource*) secara bersama sama dan mudah, menyediakan jaringan akses di mana-mana, dapat dikonfigurasi, dan layanan yang digunakan sesuai keperluan (*on demand*)[3]..

2.2.1 Karakteristik *Cloud Computing*

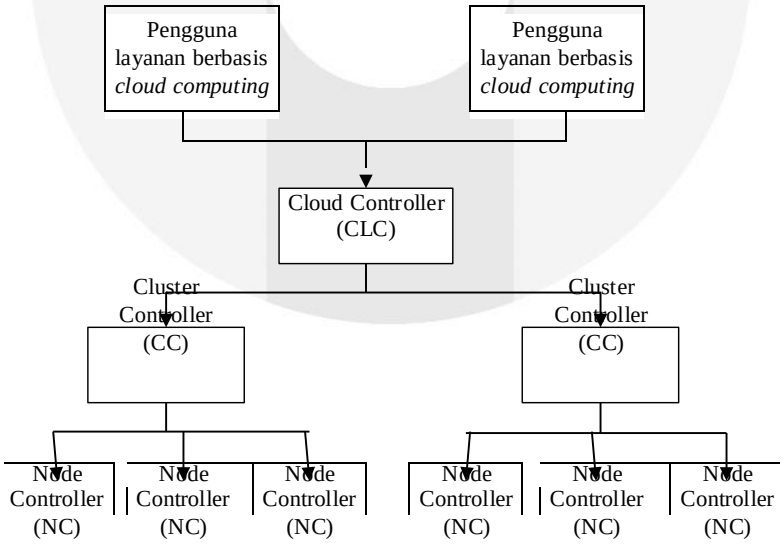
Terdapat lima buah karakteristik khusus yang dimiliki oleh *cloud computing*. Kelima karakteristik tersebut meliputi *On Demand Self Service, Broad Network Access, Resource Pooling, Rapid Elasticity dan Measured Service[6]*.

1. *On Demand Self Service*
   Merupakan karakteristik *cloud computing* dimana pengguna layanan *cloud* dapat secara mandiri menyediakan semua keperluan dan kapabilitas terkait dengan komputasi pada *cloud computing*.
2. *Broad Network Access*
   Merupakan karakteristik *cloud computing* dimana layanan *cloud* memerlukan akses jaringan komnputer yang memadai, baik pada internet, intranet, atau kombinasi keduanya.
3. Resource Pooling
   Merupakan karakteristik *cloud computing* dimana sumber daya (*resource*) komputasi dapat diberdayakan secara bersama-sama dengan lokasi fisik yang berbeda-beda.
4. Rapid Elasticity
   Merupakan karakteristik *cloud computing* dimana terjadi elastisitas yang cepat pada layanan *cloud* sesuai dengan kebutuhan pengguna yang bersifat *on demand* (sesuai dengan kebutuhan *user* sebagai pengguna layanan).
5. Measured Service
   Merupakan karakteristik *cloud computing* dimana layanan pada cloud dapat diukur. Pengukuran dapat likakukan melalui QoS(*Quality of Service*) dan QoE(*Quality of Experience*).Qos dilihat dari kualitas penyedia layanan cloud dan QoE dilihat dari pengguna layanan tersebut.

2.2.2 Layanan *Cloud Computing*

Pada cloud computing terdapat tiga model layanan yang dapat dipilih sesuai dengan kebutuhan user. Ketiga model ini meliputi IAAS, PAAS dan SAAS[6].

2.2.3 Arsitektur *Cloud Computing*



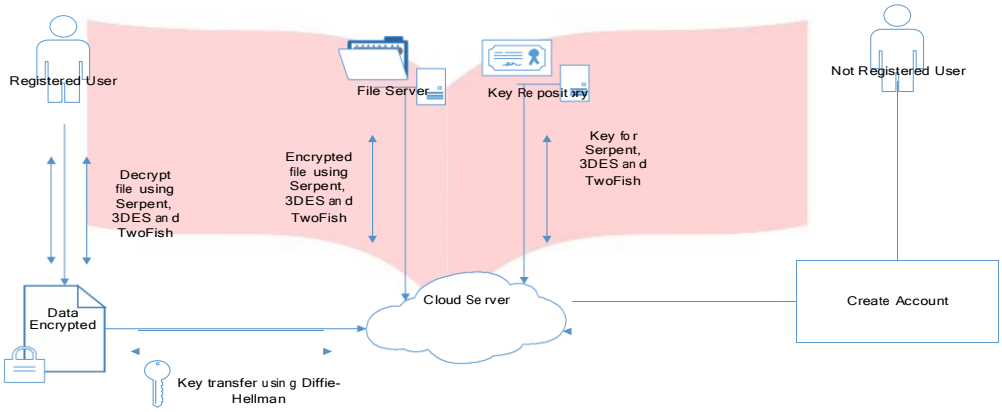
Gambar 2.3 Bagan arsitektur cloud computing secara umum[3]

Berdasarkan gambar di atas, *cloud computing* adalah sebuah layanan yang memiliki minimal sebuah *cloud controller* (CLC) yang menjadi pintu gerbang antara sistem pada *cloud computing* dengan para penggunanya. Di dalamnya terdapat satu atau beberapa buah *cluster controller*(CC), yang mana masing- masing cluster controller memiliki *node controller*(NC) di dalamnya[3].

3. Perancangan Sistem

3.1 Gambaran Umum Sistem

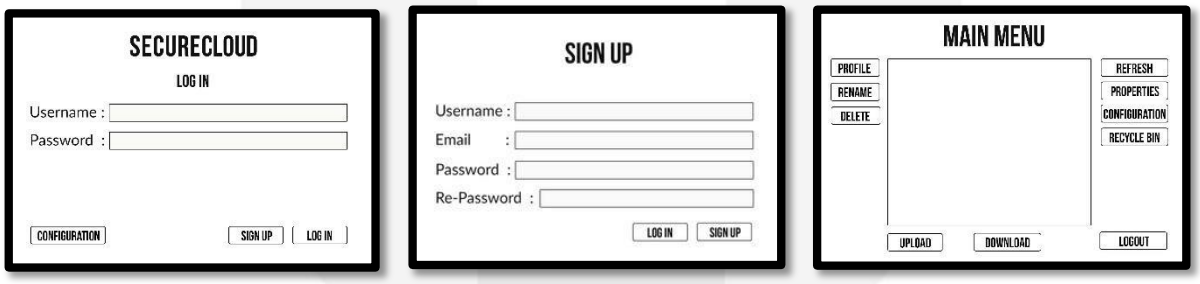
Gambaran umum sistem secara keseluruhan dapat dilihat pada gambar berikut :



Gambar 3.1 Gambaran Sistem Secara Umum [7].

3.2 Perancangan Antarmuka

Perancangan antarmuka dari sistem yang dibangun terdiri dari :



Gambar 3.2 Rancangan Tampilan Antarmuka.

4. Implementasi dan Pengujian Sistem

4.1 Implementasi Sistem

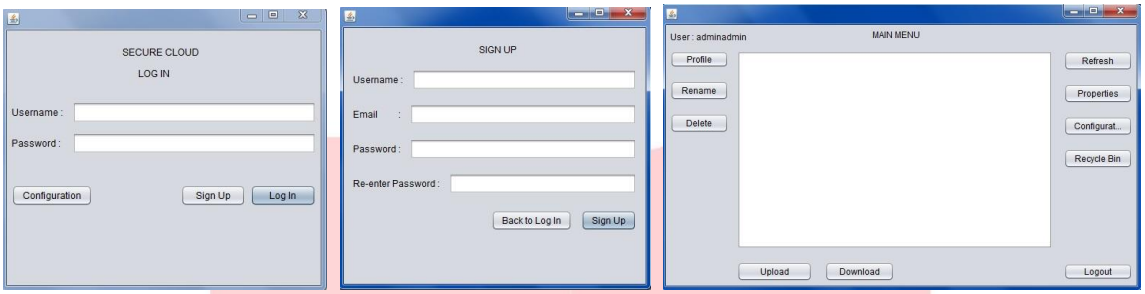
Dalam implementasi sistem enkripsi dan dekripsi *file*, langkah – langkah yang dilakukan ialah sebagai berikut:

1. Masuk kedalam Aplikasi SecureCloud dengan *Log In* atau *Sign Up* terlebih dahulu.
2. Memilih *file* dokumen yang ingin di *upload* ke dalam SecureCloud
3. Saat memilih *file* dokumen *user* dapat menganti nama *file* tersebut
4. Saat mengunggah *file* dokumen maka secara langsung *file* tersebut terenkripsi
5. Saat mengunduh *file* dokumen maka secara langsung *file* tersebut terdekripsi
6. Menghitung waktu proses enkripsi dan dekripsi dari *file* dokumen tersebut

7. Menghitung nilai *avalanche effect* data biner dari *file* dokumen tersebut
8. Menganalisis pemakain daya yang terjadi saat proses enkripsi dan dekripsi dari *file* dokumen tersebut

4.2 Implementasi Antarmuka

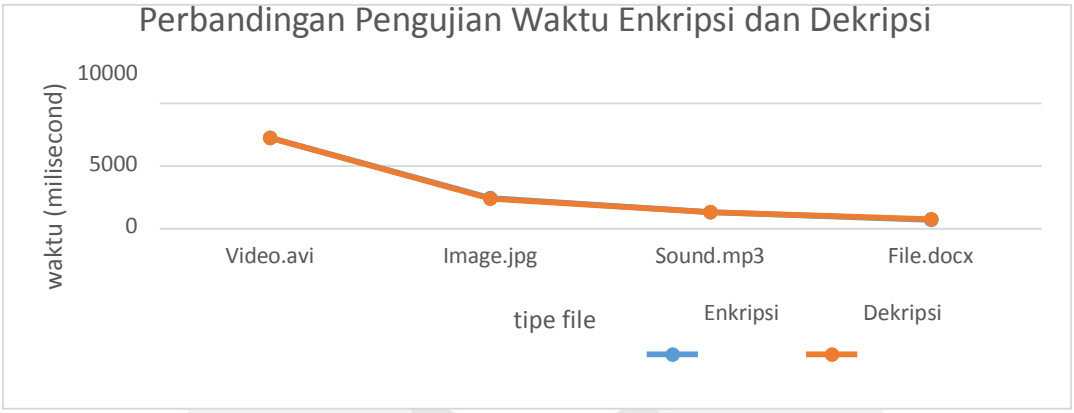
Implementasi antarmuka / *User Interface* merupakan tampilan dari aplikasi *file* yang akan menampilkan data *file* dokumen terenkripsi dan terdekripsi. Berikut merupakan tampilan antarmuka yang telah diimplementasikan :



Gambar 4.1 Implementasi Rancangan Antarmuka.

4.3 Pengujian Sistem

4.3.1 Pengujian Waktu Enkripsi dan Dekripsi



Gambar 4.2 Diagram pengujian waktu enkripsi dan waktu dekripsi

Pada pengujian waktu proses enkripsi dan waktu proses dekripsi didapatkan hasil bahwa besar ukuran dari *file* yang diuji memiliki pengaruh dalam lama waktu proses enkripsi maupun lama waktu proses dekripsi. Untuk *file* video.avi memiliki waktu proses enkripsi terlama yaitu 7270.233 *ms* dan juga waktu proses dekripsi terlama yaitu 7255.633 *ms*. Sedangkan file.docx memiliki waktu enkripsi tercepat yaitu 758.6 ms dan waktu dekripsi paling cepat yaitu 766.3 *ms*. Sistem melakukan proses enkripsi dan dekripsi *file byte* per *byte* sehingga ukuran dari *file* sangat mempengaruhi lama waktu proses enkripsi dan juga enkripsi. Jadi semakin besar ukuran *file* maka waktu yang dibutuhkan untuk proses enkripsi dan dekripsi akan lebih lama.

4.3.2 Pengujian Keamanan Sistem

Tabel 4.1 Tabel pengujian keamanan sistem

No	Nama File	Jumlah Bit Beda	Avalanche Effect %
1	File.docx	405791	50.0404 %
2	Video.avi	4117673	50.0062 %
3	Sound.mp3	742095.4	50.0140 %
4	Image.jpg	1315202	50.2781 %
Total Rata-Rata Avalanche Effect			**49.9877 %**

Pada pengujian keamanan sistem atau *Avalanche Effect* didapatkan hasil bahwa nilai AE yang tertinggi didapat di Image.jpg dengan nilai 50.2781% dan nilai AE terendah didapat di file.docx dengan nilai 50.0404%. total rata-rata AE yang didapatkan adalah 50.08% ini sudah tergolong cukup aman karena hampir mendekati lima puluh persen dari seratus persen dari total bit yang diujikan.

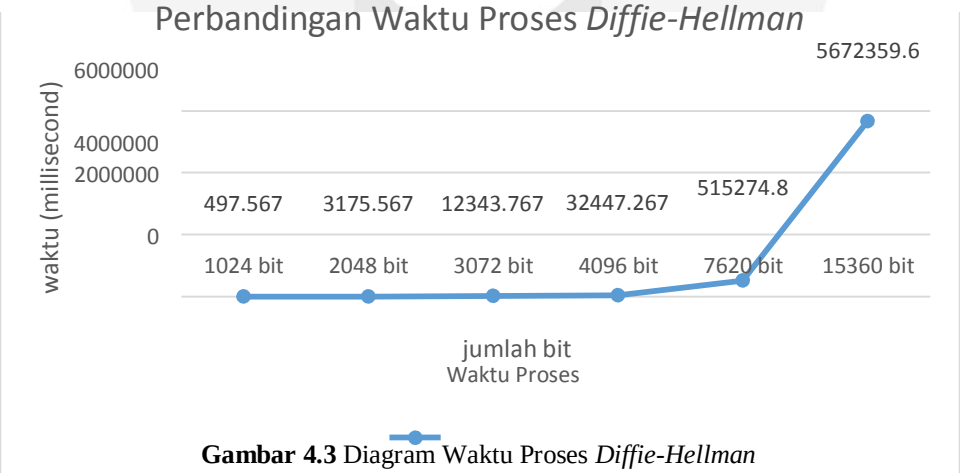
4.3.3 Pengujian Resources

Tabel 4.2 Tabel pengujian *resources*

No	Nama File	HS Enkripsi (Mega Byte)	HS Dekripsi (Mega Byte)	UH Enkripsi (Mega Byte)	UH Dekripsi (Mega Byte)
1	File.docx	59.75	58.05	14.558	8.113
2	Video.avi	67.75	74.65	22.237	28.075
3	Sound.mp3	76.867	229.917	18.301	152.187
4	Image.jpg	344.083	61.850	275.095	11.32
Total Resources		548.45	424.467	330.191	199.695

Pada pengujian *resources* ini dilakukan dengan menguji daya pemakaian dengan menggunakan *Heap Size* dan *Used Heap*. *Heap Size* adalah pengaplikasian sejumlah memori untuk aplikasi *Java Virtual Machine*. *Used Heap* adalah jumlah memori yang sesungguhnya dipakai oleh proses aplikasi. Dari hasil pengujian didapatkan nilai HS enkripsi paling rendah adalah file.docx 59.75 MB dan nilai HS dekripsi paling rendah adalah file.docx dengan 58.05 MB. Untuk nilai UH enkripsi paling rendah adalah video.avi dengan 25.167 MB dan nilai UH dekripsi paling rendah adalah file.docx dengan 12.574MB. Perbandingan nilai HS dan nilai UH pada pengujian menunjukan tingkat pemakaian daya saat aplikasi di jalankan. Semakin nilai dari HS maupun UH rendah maka daya yang digunakan untuk pemrosesan semakin kecil.

4.3.4 Pengujian Diffie-Hellman



Gambar 4.3 Diagram Waktu Proses *Diffie-Hellman*

Pada pengujian waktu proses *Diffie-Hellmann* di ketahui total rata-rata waktu proses selama 497.567 ms untuk 1024 bit, 3175.567 ms untuk 2048 bit, 12343.767 ms untuk 3072 bit, 32447.267 ms untuk 4096 bit, 515274.8 ms untuk 7620 bit dan 5672359.6 ms untuk 15360 bit. Hasil diatas menujukan bahwa dengan menggunakan *Diffie-Hellman* dengan kunci 1024 bit maka keamanannya setara dengan algoritma *3DES* 2 kunci, jika *Diffie-Hellman* dengan kunci 2048 bit maka keamanannya setara dengan *3DES* 3 kunci, jika *Diffie-Hellman* dengan kunci 3072 bit maka keamanannya setara dengan *AES*-128 bit, jika *Diffie-Hellman* dengan kunci 7680 bit maka keamanannya setara dengan *AES*-182 bit [1].

5. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian Tugas Akhir ini adalah :

1. Berdasarkan hasil pengujian yang telah dilakukan waktu ekripsi membutuhkan waktu yang sedikit lebih lama daripada waktu dekripsi. Waktu proses file di enkripsi dan di dekripsi dipengaruhi oleh besarnya ukuran file. Jadi semakin besar ukuran *file* maka waktu yang dibutuhkan untuk proses enkripsi/dekripsi akan lebih lama.
2. Pada pengujian *avalanche effect* dengan 20 file yang berekstensi tidak sama di dapatkan total nilai rata-rata *avalanche effect* adalah 50.09%. Nilai yang di dapatkan dari pengujian ini termasuk aman karena hampir mendekati lima puluh persen dari seluruh total bit yang di ujikan.
3. Berdasarkan pada pengujian *resources* didapatkan hasil 109.190 MB untuk *heap size* enkripsi dan 88.082 MB untuk *head size* dekripsi, diperoleh juga hasil 53.028 MB untuk *used heap* enkripsi dan 38.846 MB untuk *used heap* dekripsi.Semakin besar ukuran *file* yang di proses mengakibatkan nilai *heap size* dan *used heap* semakin besar.
4. Berdasakan pengujian waktu proses *Diffie-Hellman* didapatkan hasil total rata-rata waktu proses 497.567 ms untuk 1024 bit, 3175.567 ms untuk 2048 bit, 12343.767 ms untuk 3072 bit, 32447.267 *ms* untuk 4096 bit, 515274.8 *ms* untuk 7620 bit dan 5672359.6 ms untuk 15360 bit.

Daftar Pustaka

[1] Barker, E., Barker, W., Burr, W., Pork, W., & Smid, M. (2012, July). Recommendation for Key Management - Part 1 : General (Revision 3). *NIST Special Publication 800-57*, 64.

[2] Hidayat, A. (2009). ENKRIPSI DAN DEKRIPSI DATA DENGAN ALGORITMA 3 DES (TRIPLE DATA ENCRYPTION STANDARD).

[3] I Putu Agus Eka Pratama, S. M. (2014). *Smart City Beserta Cloid Computing dan Teknologi-teknologi Pendukung Lainnya.* Bandung: Informatika.

[4] Munir, R. (2006). *Kriptografi.* Bandung.

[5] O., k. S., F., I., & O., A. (2011). Cloud Computing Security Issues and Challenges. *International Journal of Computer Network (IJCN), 3*, 248.

[6] Peter Mell, T. G. (2011). The NIST Definition of Cloud Computing. *NIST Special Publication 800-145*.

[7] Tirthani, N., & R, G. (2014). Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography. *International Association for Cryptology Riset (IACR)*.

[8] W., D., & Hellman, M. (1976). "New directions in cryptography". *IEEE Transactions on Information Theory*.

[9] Wahyuni, A. (n.d.). Keamanan Pertukaran Kunci Kriptografi dengan Algoritma Hybrid : Diffie-Hellman dan RSA.