ANALISIS PERBANDINGAN ALGORITMA KRIPTOGRAFI DATA ENCRYPTION STANDARD DAN ADVANCED ENCRYPTION STANDARD PADA PROSES ENKRIPSI DAN DEKRIPSI FILE DOKUMEN

SKRIPSI

Oleh :

CARLES ANDI SAPUTRA G
NPM. 18010011

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU

2023

ENCRYPTION STANDARD PADA PROSES ENKRIPSI DAN DEKRIPSI FILE DOKUMEN

SKRIPSI

Disusun Oleh :

CARLES ANDI SAPUTRA G
NPM. 18010011

Telah Dipertahankan di depan TIM Penguji
Universitas Dehasen Bengkulu

Hari : Selasa
Tanggal : 31 Januari 2023
Tempat : Ruang Sidang/Ujian Gedung Universitas Dehasen Bengkulu

Skripsi ini telah diperiksa dan disetujui oleh TIM Penguji.

Penguji	Nama	NIDN	Tanda Tangan
Ketua	Juju Jumadi, S.Kom, M.Kom	02.111282.01	
Anggota	Eko Prasetyo Rohmawan, S.Kom, M.Kom	02.130488.01	
Anggota	Khairil, S.Kom, M.Kom	02.130475.01	
Anggota	Eko Suryana, S.Kom, M.Kom	02.151174.01	

Mengetahui,
Dekan Fakultas Ilmu Komputer

Siswanto, S.E, S.Kom, M.Kom
NIDN. 02.240363.01

ABSTRAK

ANALISIS PERBANDINGAN ALGORITMA KRIPTOGRAFI DATA ENCRYPTION STANDARD DAN ADVANCED ENCRYPTION STANDARD PADA PROSES ENKRIPSI DAN DEKRIPSI FILE DOKUMEN

Oleh :

Carles Andi Saputra G[1]
Juju Jumadi, S.Kom, M.Kom[2]
Eko Prasetyo Rohmawan, S.Kom, M.Kom[2]

Penyadapan informasi yang rahasia pun sering terjadi, sehingga perlu adanya perlindungan pada informasi tersebut, salah satunya yaitu dengan Kriptografi. Kriptografi merupakan suatu penyandian untuk melindungi informasi dengan algoritma tertentu, sehingga informasi yang rahasia dapat terlindungi. Algoritma Data Encryption Standard (DES) dan Advanced Encryption Standard (AES) yang merupakan algoritma Kriptografi Modern. Kedua algoritma ini dipilih karena sama-sama menggunakan kunci simetris dalam proses enkripsi dan dekripsi serta tergolong jenis cipher blok. Perbandingan kedua algoritma tersebut dilakukan dengan menggunakan file dokumen dengan ektensi file *.docx, *.excel, *.pdf, *.wav, *.mp4, yang masing-masing terdiri dari 5 (lima) file. File yang telah disiapkan akan diuji berdasarkan aspek perbandingan waktu proses, ukuran file dan memori yang digunakan pada saat enkripsi dan dekripsi dilakukan.

Hasil analisa kedua algoritma dapat disimpulkkan bahwa algoritma AES lebih cepat memproses enkripsi dan dekripsi dibanding DES, untuk ukuran file algoritma AES memiliki perubahan yang lebih banyak dibandingkan dengan algoritma DES dan untuk memory yang digunakan dalam proses enkripsi Algoritma AES 53.60% dan Algoritma DES 46.40%, Memory yang digunakan dalam proses dekripsi Algoritma AES 5.47% dan Algoritma DES 46.43%

Kata kunci : Kriptografi, AES, DES, Keamanan File,

1. Mahasiswa
2. Pembimbing

ABSTRACT

A COMPARATIVE ANALYSIS OF DATA ENCRYPTION STANDARD AND ADVANCED ENCRYPTION STANDARD CRYPTOGRAPHIC ALGORITHMS ON THE ENCRYPTION PROCESS AND DOCUMENT FILE DECRYPTION

By:
Carles Andi Saputra G[1]
Juju Jumadi[2]
Eko Prasetyo Rohmawan[2]

*Wiretapping of confidential information often occurs, so it is necessary to protect this information, one of which is cryptography. Cryptography is an encoding to protect information with a certain algorithm, so that confidential information can be protected. The Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithms are modern cryptographic algorithms. These two algorithms were chosen because they both use symmetric keys in the encryption and decryption process and are classified as block cipher types. Comparison of the two algorithms is done by using document files with the file extension *.docx, *.excel, *.pdf, *.wav, *.mp4, each of which consists of 5 (five) files. Files that have been prepared will be tested based on aspects of comparison of processing time, file size and memory used when encryption and decryption are performed. The results of the analysis of the two algorithms can be concluded that the AES algorithm processes encryption and decryption faster than DES, for file size the AES algorithm has more changes compared to the DES algorithm and for the memory used in the encryption process the AES algorithm is 53.60% and DES algorithm is 46.40%, The memory used in the decryption process of AES Algorithm is 5.47% and the DES Algorithm is 46.43%.*

Keywords : Cryptography, AES, DES, File Security.

1. Student
2. Supervisors

SURAT PERNYATAAN ORISINILITAS DAN PERSETUJUAN

AKADEMIK SKRIPSI

Yang bertanda tangan dibawah ini :

Nama	: Carles Andi Saputra G
NPM	: 18010011
Program Studi	: Informatika
Fakultas	: Ilmu Komputer

Dengan ini menyatakan dengan sesungguhnya bahwa Skripsi dengan Judul :

ANALISIS PERBANDINGAN ALGORITMA KRIPTOGRAFI DATA ENCRYPTION STANDARD DAN ADVANCED ENCRYPTION STANDARD PADA PROSES ENKRIPSI DAN DEKRIPSI FILE DOKUMEN

1. Adalah benar dibuat oleh saya sendiri untuk memenuhi persyaratan kelulusan akademik.
2. Pada bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah ditulis sumbernya secara jelas sesuai dengan norma, kaidah dan cara penulisan ilmiah.
3. Jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata skripsi tersebut dibuat oleh orang lain atau diketahui bahwa skripsi tersebut merupakan *plagiat/mencontek/menjiblak* hasil karya tulis ilmiah orang lain, maka dengan inisaya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku.
4. Dan atas orisinilitas tersebut diatas, maka saya menyetujui untuk memberi kepada Universitas Dehasen Bengkulu hak atas bebas royalti non-eksklusif untuk menyimpan, mengalih mediakan, mendistribusikan dan mempublikasikan skripsi saya tanpa perlu meminta izin selama mencantumkan nama saya sebagai penulis/pencipta.
5. Saya bersedia menanggung secara pribadi tanpa melibatkan Universitas Dehasen Bengkulu segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam Karya Ilmiah saya ini

Demikian surat pernyataan ini dibuat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Bengkulu, Januari 2023
Hormat Saya

Carles Andi Saputra G

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa berkat Rahmat, Hidayah, dan Karunia-Nya kepada kita semua sehingga penulis dapat menyelesaikan skripsi ini dengan judul **Analisis Perbandingan Algoritma Kriptografi DES dan AES Pada Proses Enkripsi dan Dekripsi File Dokumen**. Proposal kripsi ini disusun sebagai persyaratan dalam menyusun skripsi pada Program Studi Informatika Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.

Penulis menyadari dalam penyusunan skripsi ini tidak akan selesai tanpa bantuan dari berbagai pihak. Karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Bapak Siswanto, SE., S.Kom., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
2. Ibu Liza Yulianti, S.Kom., M.Kom selaku Ketua Program Studi Informatika Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
3. Bapak Juju Jumadi, S.Kom., M.Kom selaku Dosen Pembimbing I yang telah memberikan masukan serta arahan yang membangun dalam pembuatan skripsi ini.
4. Bapak Eko Prasetyo Rohmawan, S.Kom., M.Kom selaku Dosen Pembimbing II yang telah memberikan masukan serta arahan yang membangun dalam pembuatan skripsi ini.
5. Berbagai pihak yang telah banyak membantu dalam pembuatan proposal skripsi ini.

Penulis juga menyadari sepenuhnya bahwa di dalam skripsi ini terdapat kekurangan dan jauh dari kata sempurna. Oleh sebab itu, kami berharap adanya kritik, saran dan usulan demi perbaikan skripsi yang telah kami buat di masa yang akan datang, mengingat tidak ada sesuatu yang sempurna tanpa saran yang membangun.

Bengkulu, Januari 2023

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR LAMPIRAN

Lampiran

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan era digital saat ini membuat keamanan suatu informasi adalah suatu hal yang sangat penting. Penyadapan informasi yang rahasia pun sering terjadi, sehingga perlu adanya perlindungan pada informasi tersebut, salah satunya yaitu dengan Kriptografi. Kriptografi merupakan suatu penyandian untuk melindungi informasi dengan algoritma tertentu, sehingga informasi yang rahasia dapat terlindungi.

Berdasarkan kunci penyandiannya, kriptografi modern dibagi menjadi dua jenis yaitu kriptografi kunci simetri dan kriptografi kunci asimetri. Kriptografi kunci simetris yaitu enkripsi dan dekripsi menggunakan kunci yang sama, sedangkan kriptografi kunci asimetris yaitu menggunakan kunci yang berbeda (pasangan kunci) untuk keperluan proses enkripsi dan proses dekripsi. Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Namun, yang menjadi permasalahan dalam memilih metode kriptografi yang cocok adalah bagaimana mengetahui dan memahami cara kerja dari metode kriptografi tersebut (Purnama, 2017).

Dalam penelitian ini dilakukan kajian terhadap Algoritma Data Encryption Standard (DES) dan Advanced Encryption Standard (AES) yang merupakan algoritma Kriptografi Modern. Kedua algoritma ini dipilih karena sama-sama menggunakan kunci simetris dalam proses enkripsi dan dekripsi serta tergolong jenis cipher blok.

Perbandingan kedua algoritma tersebut dilakukan dengan menggunakan file dokumen dengan ektensi file *.docx, *.ppt, *.pdf, *.wav, *.mp4, yang masing-masing terdiri dari 5 (lima) file. File yang telah disiapkan akan diuji berdasarkan aspek perbandingan waktu proses, ukuran file dan memori yang digunakan pada saat enkripsi dan dekripsi dilakukan.

Penelitian terkait dilakukan oleh (Meko, 2018) dengan judul "Perbandingan Algoritma DES, AES, IDEA dan Blowfish Dalam Enkripsi dan Dekripsi Data". Penelitian ini bertujuan untuk membandingkan kinerja beberapa algoritma kriptografi dalam proses enkripsi dan dekripsi data berdasarkan segi kecepatan atau lama waktu serta ukuran file hasil enkripsi. Hasil penelitian ini menunjukkan adanya perbedaan waktu proses serta ukuran file dari hasil enkripsi dan dekripsi data dari masing-masing algoritma.

Berdasarkan uraian tersebut di atas, maka penulis tertarik untuk mengangkat judul penelitian yaitu tentang **Analisis Perbandingan Algoritma Kriptografi DES dan AES Pada Proses Enkripsi dan Dekripsi File Dokumen**.

1.2. Rumusan Masalah

Dari uraian latar belakang tersebut, maka dapat dirumuskan masalah, yaitu bagaimana menganalisis perbandingan algoritma kriptografi DES dan AES pada proses enkripsi dan dekripsi file dokumen ?

1.3. Batasan Masalah

Agar tidak melebar dari permasalahan yang akan dibahas, maka penulis membatasi masalah dalam penelitian ini, yaitu :

1) Data untuk analisis perbandingan yaitu file dokumen dengan ektensi file *.docx, *.ppt, *.pdf, *.wav, *.mp4, *.
2) Aspek perbandingan dari kedua algoritma tersebut yaitu berdasarkan waktu proses, ukuran file dan memori yang digunakan pada saat enkripsi dan dekripsi.
3) Bahasa pemrograman yang digunakan adalah Visual Basic .Net.

1.4. Tujuan Penelitian

Tujuan penelitian ini dibagi menjadi 2 (dua) bagian yaitu Tujuan Umum dan Tujuan Khusus. Adapun tujuan penelitian ini, antara lain :

1) Tujuan Umum

   Untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Informatika Universitas Dehasen Bengkulu.

2) Tujuan Khusus
   a) Untuk menganalisis perbandingan algoritma kriptografi DES dan AES pada proses enkripsi dan dekripsi file dokumen
   b) Untuk mengetahui informasi waktu proses, ukuran file, dan memori yang digunakan dalam proses enkripsi dan dekripsi file dokumen berdasarkan algoritma kriptografi DES dan AES

1.5. Manfaat Penelitian

Manfaat dari penelitian ini, antara lain :

1. Dapat memberikan sebuah solusi dalam memilih algoritma kriptografi yang terbaik berdasarkan hasil analisis perbandingan algoritma DES dan AES
2. Dapat mengetahui informasi waktu proses, ukuran file dan memori yang digunakan dalam proses enkripsi dan dekripsi file dokumen
3. Dapat membantu mengamankan file dokumen menggunakan algoritma DES dan AES.
4. Dapat dijadikan bahan referensi dalam pembuatan aplikasi kriptografi menggunakan algoritma DES dan algoritma AES.

BAB II

LANDASAN TEORI

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani, crypto dan graphia. Crypto berarti secret (rahasia) dan graphia berarti writing (tulisan). Kriptografi menurut terminologinya adalah sebuah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Secara istilah kriptografi didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan baik berupa data maupun informasi yang mempunyai arti atau nilai dengan cara menyamarkan (mengacak) menjadi bentuk yang tidak dapat dimengerti dan hanya penerima yang dapat mengubah kode-kode tersebut menjadi pesan asli yang dapat dimengerti. Kriptografi adalah ilmu mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data dan otentikasi (Jamaludin & Romindo, 2020).

Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan :

1. Kerahasiaan (*confidentiality*) adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas data (*data integrity*) adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.

3. Otentikasi (*authentication*) adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user autehentication*).
4. *Non-repudiation* adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan.

Di dalam kriptografi akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, *Plainteks*, dan *Cipherteks*

   Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (*plaintext*) atau teks jelas (*cleartext*).

2. Pengirim dan Penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.

3. Enkripsi dan deskripsi

   Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering* (standar nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula disebut deskripsi (*decryption*) atau *deciphering* (standar nama menurut ISO 7498-2).

4. *Cipher* dan kunci

   Algoritma kriptogarfi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan

dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plainteks* dan himpunan yang berisi *cipherteks*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan *plainteks* dan C menyatakan *cipherteks*, maka :

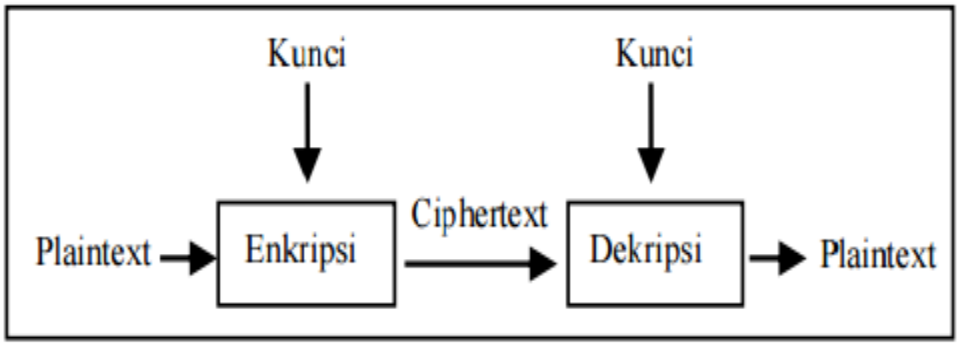
E(P) = C → fungsi enkripsi E memetakan P ke C

D(C) = P → fungsi dekripsi D memetakan C ke P

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar.

Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa *string* atau deretan bilangan.

Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai skema diperlihatkan pada Gambar 2.1.

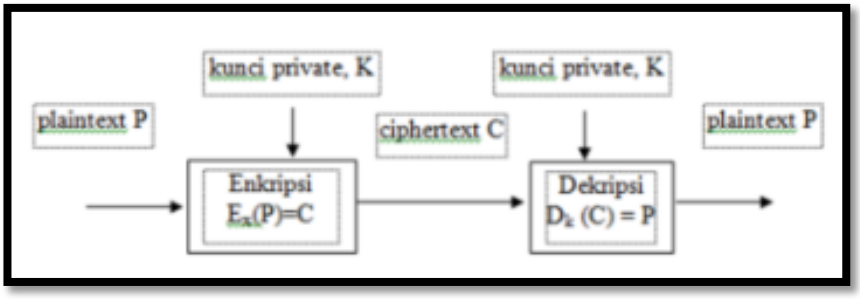


Gambar 2.1. Skema Enkripsi dan Dekripsi Dengan Menggunakan Kunci

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi kriptografi kunci simetris (symmetric-key cryptography) dan kriptografi kunci asimetris (asymmetric-key cryptography) (Sari, et al., 2020) :

1) Kriptografi Kunci Simetris

   Pada sistem kriptografi kunci simetris, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itu dinamakan kriptografi kunci simetris atau kriptografi simetris. Istilah lain untuk kriptografi simetris adalah kriptografi kunci privat (private key cryptography), kriptografi kunci rahasia (secret key cryptography) atau kriptografi konvensional (conventional cryptography). Sistem kriptografi simetris mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan kriptografi simetris terletak pada kerahasiaan kuncinya. Kriptografi simetris merupakan satu-satunya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976. Semua kriptografi klasik termasuk ke dalam sistem kriptografi simetris. Di sisi lain, ada puluhan algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi simetris, diantaranya adalah DES (Data Encryption Standard), Blowfish, Twofish, Triple-DES, IDEA, Serpent, dan yang terbaru adalah AES (Advanced Encryption Standard). Skema kriptografi kunci simetris seperti Gambar 2.2.

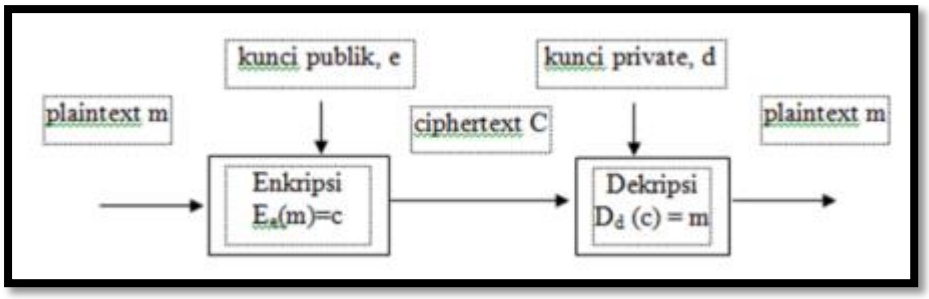


Gambar 2.2. Skema Kriptografi Kunci Simetris

2) Kriptografi Kunci Asimetris

Jika kunci untuk enkripsi tidak sama dengan kunci untuk dekripsi, maka kriptogafinya dinamakan kriptografi kunci asimetris atau kriptografi asimetris. Nama lainnya adalah kriptografi kunci publik (public key cryptography), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun (diumumkan ke publik), sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan. Pada kriptografi jenis ini, setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim mengenkripsi pesan dengan menggunakan kunci publik si penerima pesan. Hanya penerima pesan yang dapat mendeskripsikan pesan karena hanya dia yang mengetahui kunci privatnya sendiri. Untuk berkomunikasi secara rahasia dengan banyak orang, tidak perlu kunci rahasia sebanyak jumlah orang tersebut, cukup membuat dua buah kunci yaitu kunci publik bagi para koresponden untuk mengenkripsikan pesan dan kunci privat bagi penerima pesan untuk mendeskripsikan pesan. Berbeda dengan kunci simetris dimana jumlah kunci yang dibuat adalah sebanyak jumlah pihak yang diajak untuk berkorespondensi. Contoh algoritma kriptografi kunci publik diantaranya RSA, Elgamal, DSA, Knapsack, Elliptic Curve dan

lain sebagainya. Adapun skema kriptografi kunci asimetris seperti Gambar 2.3.

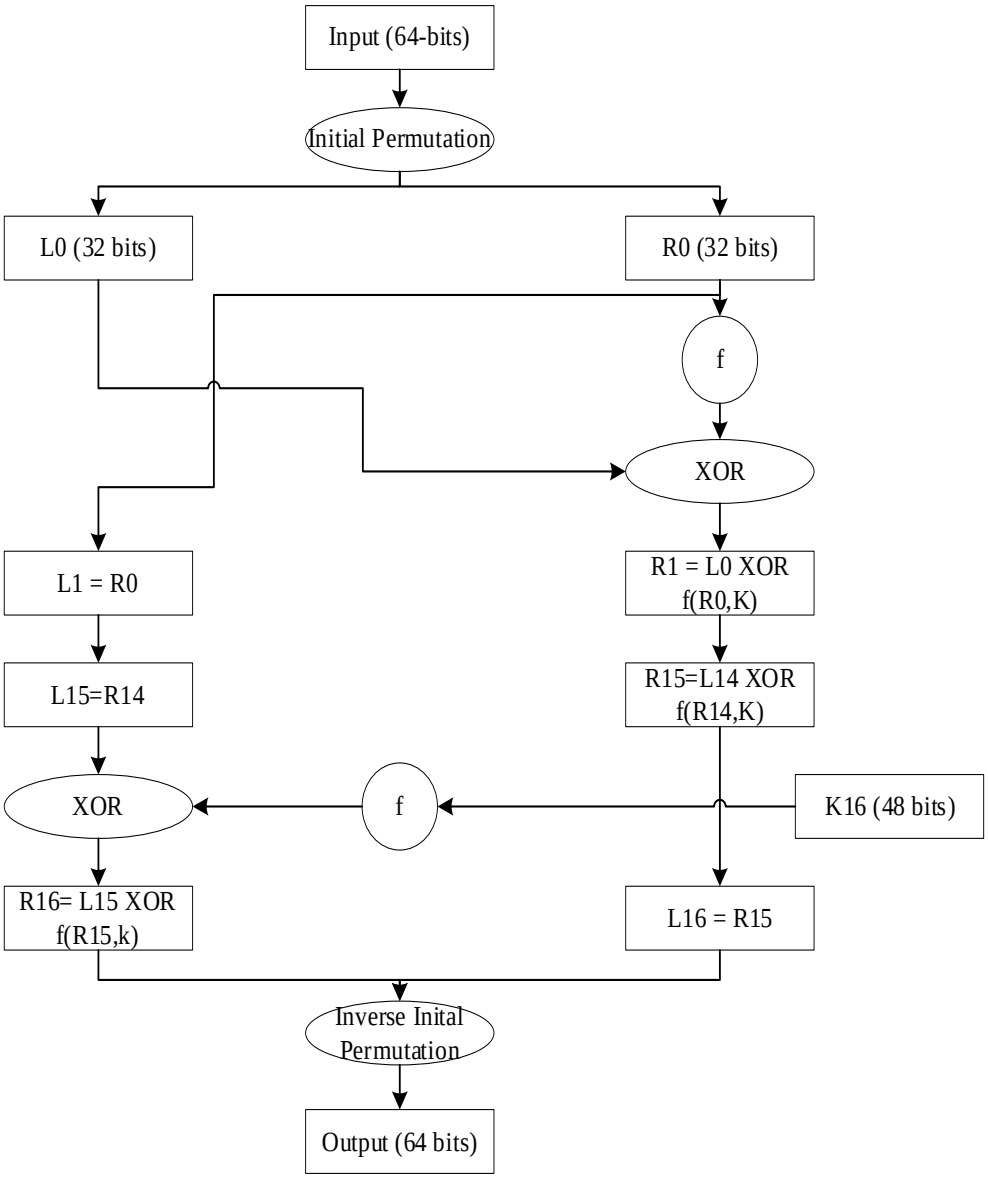


Gambar 2.3. Skema Kriptografi Kunci Asimetris

2.2. Algoritma DES

DES, atau juga dikenal sebagai *Data Encryption Algorithm* (DEA) oleh ANSI dan DEA-1 oleh ISO, merupakan algoritma kriptografi simetris yang paling umum digunakan saat ini. Secara umum, DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. 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 lupa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit (Meko, 2018).

Penggunaan data sandi yang paling banyak didasarkan pada *standard-standard* data sandi (DES) yang diambil pada tahun 1977 oleh Standard-Standard Nasional Bureau, yang sekarang Institut Nasional Standard dan Teknologi (NIST), sebagai Standard Proses Informasi Umum. Untuk DES, data disandikan ke dalam 64 balok bit menggunakan 56 bit kunci. Transformasi algoritma 64 bit input ke dalam satu seri langkah-langkah ke dalam 64 bit output. Langkah yang sama dengan kunci yang sama, digunakan untuk cadangan persandian (Adhar, 2019).



Gambar 2.4. Flowchart Algoritma DES

Komponen pada sandi *Fiestel* adalah memanfaatkan sandi produksi yang digunakan secara berulang dalam beberapa ronde. Komponen pada sandi *Fiestel* dapat bersifat *self-invertible* (*invers* dengan komponen yang sama), *invertible* (memiliki *invers*) *non-invertible* (tidak memiliki *invers*) (Rifki Sadikin, 2012). DES merupakan salah satu contoh sandi *Fiestel*, sehingga struktur sandi DES memiliki struktur yang sama dengan sandi *Fiestel* dengan penyusunan. Panjang blok DES adalah 64 bit, jadi ukuran

teks asli dan teks sandi adalah 64 bit. Ukuran kunci DES adalah 64 bit dan ukuran kunci ronde adalah 48 bit dan jumlah ronde pada DES adalah 16 ronde. Struktur sandi DES dalam skema global adalah sebagai berikut :

1. Blok *plainteks* di permutasi 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 di permutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP-1) menjadi blok *cipherteks*.

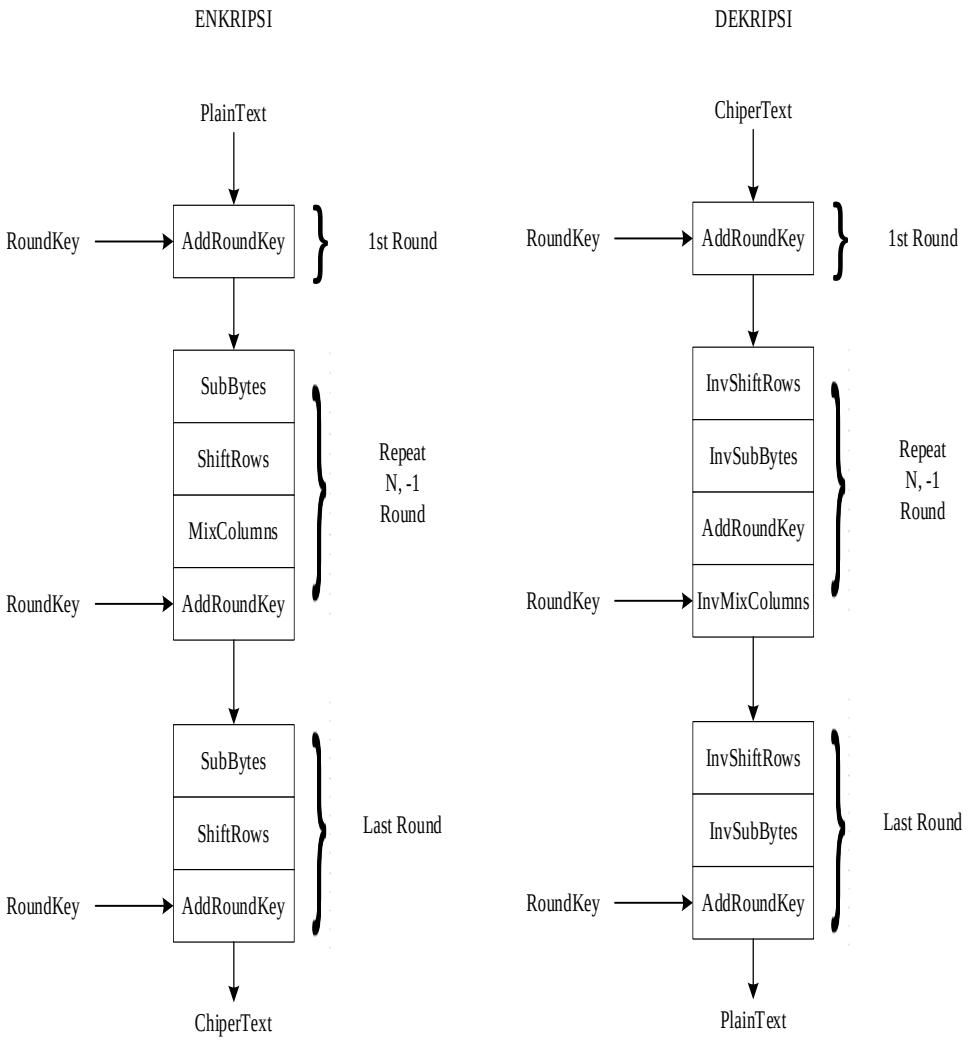
Enkripsi dan dekripsi DES memiliki 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K1,K2,…,K16. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter.

Suatu kunci 64 bit digunakan sebagai input kunci akan tetapi tiap bit kedelapan diabaikan akibatnya menghasilkan 56 bit input dan disajikan dalam dua bit *string* yang berjumlah masing-masing 28 bit dan selanjutnya mengalami transformasi *left shifts* dimana setiap *Ki round* dilakukan perputaran 1 atau 2 bit sesuai dengan *round Ki*. Proses enkripsi terhadap blok *plainteks* dilakukan setelah permutasi awal (IP). Tujuan permutasi awal adalah mengacak *plainteks* sehingga urutan bit-bit di dalamnya berubah.

2.3. Algoritma AES

Algoritma AES adalah algoritma kritografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi,

yaitu 128 bit, 192 bit, dan 256 bit. Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*. Awal proses enkripsi, input yang telah disalin ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, state akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, *dan AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut dengan *round function*. *Round* terakhir agak berbeda dengan *round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns* (Handoyo & Subakti, 2020).



Gambar 2.5. Flowchart Algoritma AES

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok *chipertext* simetris yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*, sebaliknya dekripsi adalah mengubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekripsi data. Secara umum metode yang digunakan dalam pemrosesan terbagi dua, yaitu (Permana & Nurnaningsih, 2018)

1. Enkripsi

   Enkripsi adalah proses penyandian *plaintext* menjadi *ciphertext*, atau pengubahan data menjadi bentuk rahasia. Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey*. Pada awal proses enkripsi, input yang telah dicopykan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows, MixColumns, dan AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*.

2. Dekripsi

   Dekripsi adalah proses memperoleh kembali *plaintext* menjadi *ciphertext*, atau proses pengubahan kembali data yang berbentuk rahasia

menjadi semula. Transformasi *byte* yang digunakan pada *invers* cipher adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*.

Pada Proses enkripsi awalnya teks asli dibentuk sebagai sebuah state. Kemudian sebelum ronde 1 dimulai blok teks asli dicampur dengan kunci ronde ke-0 (transformasi ini disebut *AddRoundKey*). Setelah itu, ronde ke-1 sampai dengan ronde ke-(Nr-1) dengan Nr adalah jumlah ronde. AES menggunakan 4 jenis transformasi yaitu:

1. *SubBytes*, sebagai transformasi substitusi.
2. *ShiftRows*, sebagai transformasi permutasi.
3. *MixColumns*, sebagai transformasi pengacakan.
4. *AddRoundKey*, sebagai transformasi penambahan kunci.

Pada ronde terakhir, yaitu ronde ke-*Nr* dilakukan transformasi serupa dengan ronde lain namun tanpa transformasi serupa dengan ronde lain namun tanpa transformasi *MixColumns*. Secara ringkas algoritma deskripsi merupakan kebalikan algoritma enkripsi AES. Algoritma dekripsi AES menggunakan transformasi *invers* semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar dari algoritma kriptografi AES memiliki transformasi *invers*, yaitu: *InvSubBytes*, *InvShiftRows* dan *InvMixColumns*. *AddroundKey* merupakan transformasi yang bersifat *self-invers* dengan syarat menggunakan kunci yang sama

Penyandian AES membutuhkan kunci ronde untuk setiap ronde transformasi kunci ronde ini di bangkitkan (di ekspansi) dari kunci AES. Pada bagian ini di bahas bagaimana kunci ronde di bangkitkan oleh kunci AES. Kunci AES 128 bit atau 4 *word* menghasilkan sebuah larik sebanyak

44 *word* yang menjadi kunci. Berikut adalah langkah-langkah mengekspansi kunci:

1. Pertama kunci AES 128 bit di organisir menjadi 4 *word* dan disalin ke *word* keluaran (W) pada 4 elemen pertama (W[0], W[1], W[2], W[3]).
2. Untuk elemen keluaran selanjutnya W[i] dengan i ={4,…,43} dihitung sebagai berikut :
   a. Salin W [i-1] pada *word* t.
   b. Jika i mod 4 = 0 (I habis dibagi 4) maka lakukan W[i]= f (t,i) ⊕ W[i-4] ,dengan fungsi f(t,i) adalah f(t,i) = Subword (rotword(t)) ⊕ RC[i/4]
   c. Jika I mod 4 tidak sama dengan 0, lakukan W[i]= t ⊕ W[i-4].

2.4. Visual Basic .Net

Microsoft Visual Basic .Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .Net *Framework*, dengan menggunakan bahasa *basic*. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi *windows form*, aplikasi web berbasis ASP.Net dan juga aplikasi *command-line*. Bahasa Visual Basic .Net sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .*Net Framework* (Blazing, 2018).

Visual Studio adalah IDE (*Integrated Development Environment*) yang dapat digunakan untuk mengembangkan aplikasi-aplikasi *Windows*. Visual studio dirancang untuk fokus pada produktivitas. Tool ini disebut

juga *Rapid Application Development Tools* (*RAD tools*) karena dirancang dan dilengkapi untuk meningkatkan produktivitas. Versi baru dari Visual Studio inversi terbaru dibuat lebih sederhana untuk mempermudah pengguna dalam mempelajarinya dan memenuhi kebutuhan para *Programmer* (Enterprise, 2015).

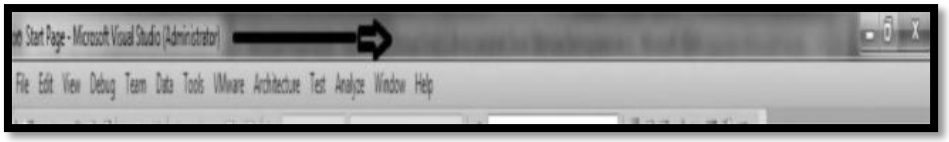
Adapun tampilan Visual Studio secara keseluruhan, seperti Gambar 2.6. (Blazing, 2018).

Gambar 2.6. Tampilan Visual Studio

Komponen yang terdapat pada Visual Studio antara lain (Blazing, 2018) :

1. Tittle Bar

   Tittle bar adalah informasi nama project yang sedang dibuat. Adapun komponen tittle bar seperti Gambar 2.7.



Gambar 2.7. Tittle Bar

2. Menu Bar

   Menu bar yang terdapat pada program-program aplikasi di Windows. Menu bar digunakan untuk melakukan proses atau perintah-perintah tertentu. Menu bar dibagi menjadi beberapa pilihan sesuai dengan kegunaannya. Adapun komponen menu bar, seperti Gambar 2.8.

Gambar 2.8. Menu Bar

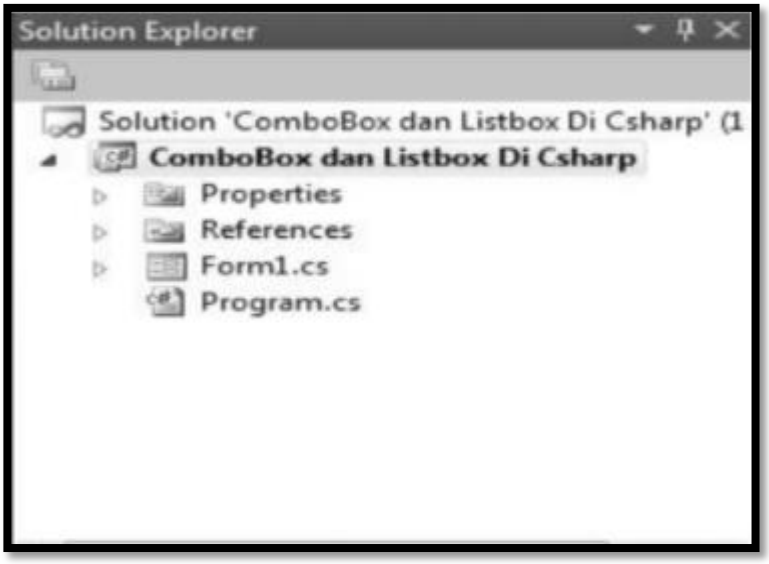
3. Toolbars

   Toolbars pada aplikasi windows lainnya yang berisi tombol-tombol yang mewakili suatu perintah tertentu yang sering digunakan untuk keperluan dalam pemrograman dan lain-lain. Adapun komponen toolbars, seperti Gambar 2.9.

Gambar 2.9. Toolbars

4. Solution Explorer

   Solution explorer adalah jendela yang menyimpan informasi mengenai solution, project-project, beserta file-file, form-form ataupun resource yang digunakan pada program aplikasi. Adapun komponen solution explorer, seperti Gambar 2.10.



Gambar 2.10. Solution Explorer

5. Toolbox

   Toolbox adalah tempat penyimpanan kontrol-kontrol atau komponen standar yang nantinya akan diletakkan sebagai komponen program di dalam form saat merancang sebuah aplikasi. Adapun komponen toolbox, seperti Gambar 2.11.

Gambar 2.11. Toolbox

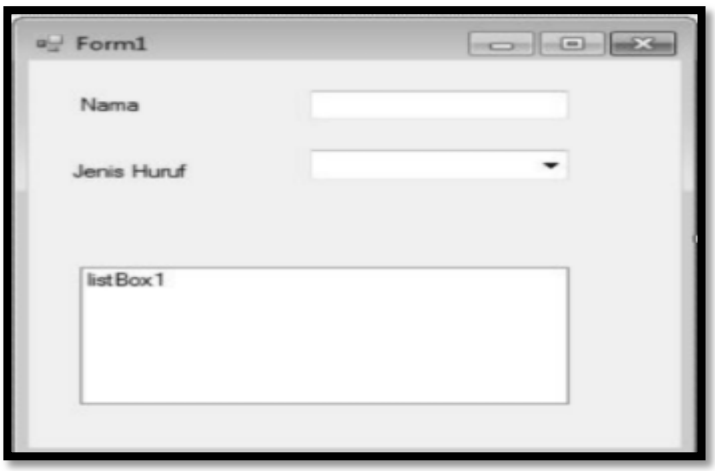
6. Properties

   Jendela properties berfungsi untuk memberikan informasi mengenai objek yang sedang aktif, nama objek yang sedang aktif dapat dilihat pada bagian atas jendela properties. Properties juga digunakan untuk mengubah nilai property atau karakteristik dari objek yang aktif. Adapun komponen properties, seperti Gambar 2.12.

Gambar 2.12. Properties

7. Form

   Form merupakan suatu objek yang digunakan untuk merancang tampilan program. Adapun komponen form, seperti Gambar 2.13.



Gambar 2.13. Form

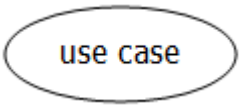
2.5. *Unified Modeling Language* (UML)

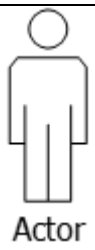
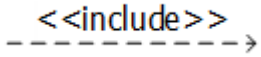
Unified Modeling Language merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa & Shalahuddin, 2016).

2.5.1. *Use Case Diagram*

Use case atau diagram use case merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol yang ada pada diagram use case.

Tabel 2.1. Use Case Diagram

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case
Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang

	akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/Association	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Generalisasi/ generalization 	Hubungan generalisasi atau spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Misalnya arah panah pengarah pada use case yang menjadi generalisasinya (umum).
Menggunakan/ include/ uses 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahakn memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

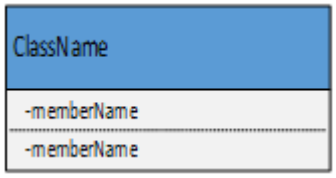
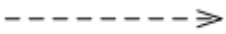
2.5.2. *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas.

Tabel 2.2. Class Diagram

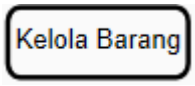
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi/association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/directed association 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/dependency 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/aggregation 	Relasi antarkelas dengan makna semua bagian (whole-part)

2.5.3. *Activity Diagram*

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan

aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

Tabel 2.3. Activity Diagram

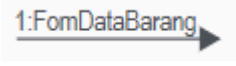
Simbol	**Keterangan**
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.5.4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakukan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk

mengambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Berikut adalah simbol-simbol yang ada pada diagram sekuen.

Tabel 2.4. Sequence Diagram

Simbol	**Keterangan**
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis Hidup/Lifeline	Menyatakan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan Tipe Call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

Pesan Tipe Create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan Tipe Destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

2.6. *Flowchart*

Flowchart adalah dalam bahasa Indonesia diagram alir, merupakan diagram yang memuat simbol-simbol grafis yang menyatakan aliran algoritma atau proses dari langkat-langkat instruksi dalam bentuk-bentuk kotak persegi dan bulat dan pernyataan instruksi, dimana hubungan dan urutan proses tiap instruksi ditujukan dengan simbol tanda panah (Anggrawan, 2018).

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso & Nurmalina, 2017).

Flowchart adalah bagan-bagan yang mempunyai arus menggambarkan langkah-langkah penyelesaian suatu masalah merupakan cara penyajian dari suatu algoritma. Ada 2 (dua) macam *flowchart* :

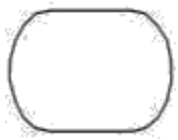
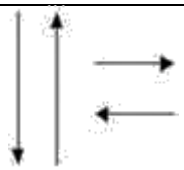
a. *System Flowchart*

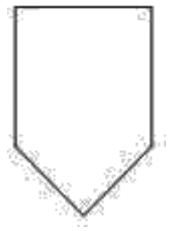
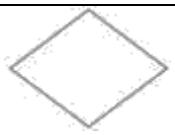
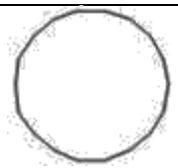
Urutan proses dalam sistem dengan menunjukkan alat media *input*, *output* serta jenis penyimpanan dalam proses pengolahan data.

b. Program *Flowchart*

Urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program.

Tabel 2.5. Simbol *Flowchart*

Simbol	Keterangan	Penjelasan
	Simbol Terminator (simbol start dan end)	Simbol untuk tanda mulai (start) dan tanda selesai (stop/end) dari kegiatan proses
	Simbol Arah Aliran	Simbol yang menghubungkan antara simbol yang satu dengan simbol lainnya (atau antara kegiatan proses) dan sekaligus menyatakan arah proses
	Simbol keluaran/masukan (Input/output)	Simbol yang menyatakan proses *input* dan *output* (berlaku untuk semua media input dan output)
	Simbol Proses	Simbol yang melambangkan kegiatan pemrosesan/pengolahan input

	Simbol Konektor	Simbol untuk tanda penyambungan proses pada lembar atau halaman yang berbeda.
	Simbol Percabangan atau Pilihan Keputusan	Simbol proses pemilihan keputusan tergantung kondisi, jika pemeriksaan kondisi terpenuhi benar maka jalur pilihan yang diproses adalah jalur ya atau yes, dan sebalinya jika pemeriksaan tidak terpenuhi tidak benar, maka jalur tidak atau No.
	Simbol Konektor	simbol untuk tanda penyambungan proses pada lembar atau halaman yang sama

BAB III

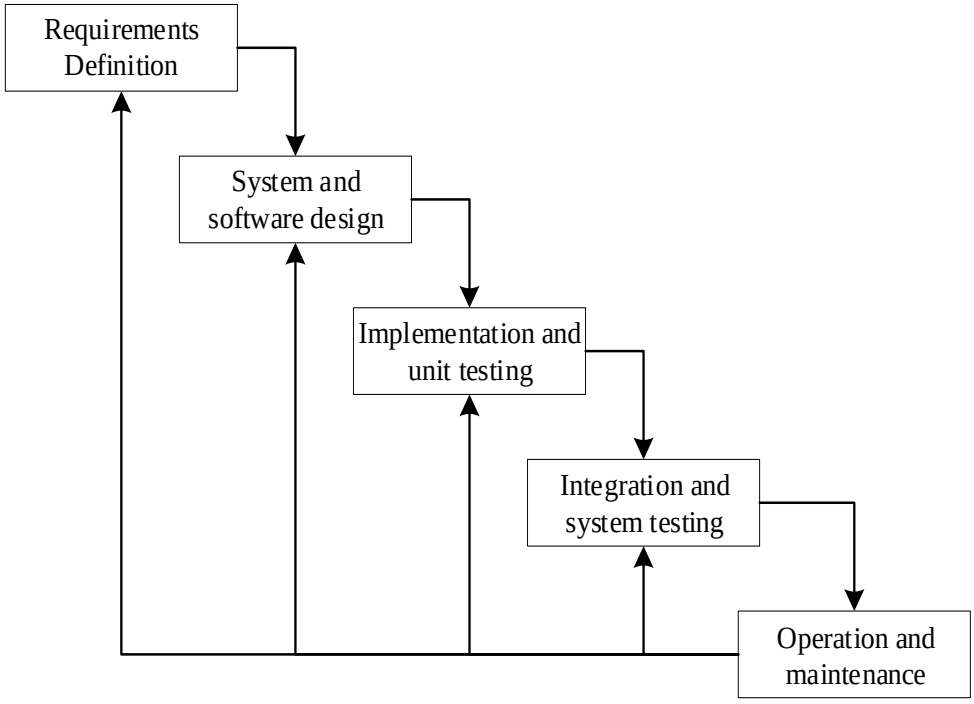
METODOLOGI PENELITIAN

3.1. Subjek Penelitian

Penelitian ini dilakukan secara mandiri dimana tidak terikat terhadap instansi atau tempat penelitian. Waktu penelitian dilakukan pada bulan April 2022 sampai dengan September 2022 (terlampir).

3.2. Metode Penelitian

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematik dan sekuensial. Metode *waterfall* memiliki tahapan-tahapan seperti Gambar 3.2.



Gambar 3.2. Metode Waterfall

Keterangan :

1) *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2) *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3) *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4) *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*

5) *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak

ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

3.3. Perangkat Keras dan Perangkat Lunak

1. Perangkat Keras (*Hardware*)

   Perangkat keras (*Hardware*) yang digunakan dalam penelitian ini, antara lain :

   a. Laptop Asus
   b. Processor Intel Inside
   c. Memory RAM 4GB
   d. Hardisk 500GB

2. Perangkat Lunak (*Software*)

   Perangkat lunak (*Software*) yang digunakan dalam penelitian ini, antara lain :

   a. Sistem Operasi Windows 10
   b. Visual Studio 2010
   c. Microsoft Visio 2010

3.4. Metode Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini, antara lain :

a. Studi Dokumen

Teknik data dengan dokumentasi adalah metode yang lebih mudah dilakukan metode-metode lain karena jika ada kekeliruan, sumber

datanya masih tetap. Objek yang diamati pada metode dokumentasi kesalahan benda hidup melainkan benda mati.

b. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan atau instansi yang berupa karya ilmiah, jurnal, buku-buku yang berhubungan dengan penulisan ini.

3.5. Metode Perancangan Sistem

3.5.1. Analisa Sistem Aktual

Banyaknya algoritma kriptografi baik klasik atau pun modern yang dibagi berdasarkan jenis kunci yang digunakan, di mana terdapat 2 jenis kunci tersebut yaitu kunci asimetris dan simetris. Setiap algoritma mempunyai spesifikasi, karakter yang berbeda-beda dan memiliki kelebihan serta kekurangan masing-masing. Namun pada dasarnya hal yang penting dalam proses enkripsi yaitu kunci dan penentuan algoritma yang dapat mempengaruhi performansi proses enkripsi sehingga informasi yang ingin dilindungi akan lebih aman.

3.5.2. Analisa Sistem Baru

Analis sistem baru dilakukan berdasarkan permasalahan yang terdapat pada sistem aktual. Dalam penelitian ini dilakukan kajian terhadap Algoritma Data Encryption Standard (DES) dan Advanced Encryption Standard (AES) yang merupakan algoritma Kriptografi Modern. Kedua algoritma ini dipilih karena sama-sama

menggunakan kunci simetris dalam proses enkripsi dan dekripsi serta tergolong jenis cipher blok. Perbandingan kedua algoritma tersebut dilakukan dengan menggunakan file dokumen dengan ektensi file *.docx, *.ppt, *.pdf, *.wav, *.mp4, *.jpeg, *.png yang masing-masing terdiri dari 5 (lima) file. File yang telah disiapkan akan diuji berdasarkan aspek perbandingan waktu proses, ukuran file dan memori yang digunakan pada saat enkripsi dan dekripsi dilakukan.

A. Penerapan Algoritma DES

Untuk pemahaman dalam penerapan Algoritma DES, maka diambil plainteks dan kunci sebagai berikut :

Plainteks = Carles

Kunci = Andi

Penyelesaian :

Tabel 3.1 Plainteks Konversi ke Biner

Plainteks	Biner
C	01000011
a	01100001
r	01110010
l	01101100
e	01100101
s	01110011

Tabel 3.2 Kunci Konversi ke Biner

Kunci	Biner
A	01000001
n	01101110
d	01100100
i	01101001

Initial permutation (IP) pada bit plaintext menggunakan tabel IP berikut :

0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	1	0
0	1	1	0	1	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

1	1	0	0	0	0	1	1
0	0	1	0	0	1	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	0
0	0	1	0	0	1	0	1
0	1	1	1	0	0	0	1
1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0

Sehingga hasil outputnya adalah :

IP(x) : 11000011 00100101 01010010 01101000 00100101 01110001 10010000 00001000

Pecah bit pada IP(x) Plaintext menjadi 2 bagian yaitu:

L0 : 11000011 00100101 01010010 01101000

R0 : 00100101 01110001 10010000 00001000

Generate kunci yang akan digunakan untuk mengenkripsi plaintext dengan menggunakan tabel permutasi :

0	1	0	0	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	0	0	1	0	0
0	1	1	0	1	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	1
0	0	1	0	1	0	1	0
0	1	0	0	0	1	0	0
0	1	1	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0

Berikut hasil outputnya :

CD(k) : 00000001 00101010 01000100 01100001 01000000 00000010 10010000 00001000

Pecah CD(k) menjadi dua bagian kiri dan kanan, sehingga menjadi

C0 : 00000001 00101010 01000100 01100001

D0 : 01000000 00000010 10010000 00001000

Melakukan pergeseran kiri (Left Shift) pada C0 dan C1 sebanyak 16 putaran sehingga diperoleh hasil :

C1 : 00000010 01010100 10001000 11000010

D1 : 10000000 00000101 00100000 00010000

C2 : 00000100 10101001 00010001 10000100

D2 : 00000000 00001010 01000000 00100000

C3 : 00001001 10100100 00100011 00001000

D3 : 00000000 00010100 10000000 01000000

C4 : 00010011 10001000 01000110 00010000

D4 : 00000000 00101001 00000000 10000000

C5 : 00100111 00100000 10001100 00100000

D5 : 00000000 01010010 00000001 00000000

C6 : 01001110 01000001 00011000 01000000

D6 : 00000000 10100100 00000010 00000000

C7 : 10011100 10000010 00110000 10000000

D7 : 00000001 01001000 00000100 00000000

C8 : 00111001 00000100 01100010 00000000

D8 : 00000101 01000000 00010000 00000000

C9 : 01110010 00001000 11000100 00000000

D9 : 00001010 10000000 00100000 00000000

C10 : 11100100 00010001 10001000 00000000

D10 : 00010101 00000000 01000000 00000000

C11 : 11001000 00100011 00010000 00000000

D11 : 00101010 00000000 10000000 00000000

C12 : 10010000 01000110 00100000 00000000

D12 : 01010100 00000001 00000000 00000000

C13 : 00100000 10001100 01000000 00000000

D13 : 10101000 00000010 00000000 00000000

C14 : 01000001 00011000 10000000 00000000

D14 : 01010000 00000100 00000000 00000000

C15 : 10000010 00110001 00000000 00000000

D15 : 10100000 00001000 00000000 00000000

C16 : 00000100 01100010 00000000 00000000

D16 : 01000000 00010000 00000000 00000000

Ekspansi data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1<= i <=16.

L0 : 11000011 00100101 01010010 01101000

R0 : 00100101 01110001 10010000 00001000

L1 = R0

L1 = 00100101 01110001 10010000 00001000

R1 = L0 XOR f(R0,K)

R1 = 11000011 00100101 01010010 01101000 XOR

00000010 01010100 10001000 11000010

R1 = 11000001 01110011 11011010 10101010

Dan seterusnya hingga iterasi 16 menggabungkan R16 dengan L16 kemudian dipermutasikan untuk terakhir kali dengan tabel Invers Initial Permutasi (IP-1).

Sehingga menghasilkan output biner :

R16L16 = 01110000 00011001 11010011 11010101 00010011 11110111 01110111 00100011

Menghasilkan Hasil Ciphertext = *f*'±œš³

B. Penerapan Algoritma AES

Untuk pemahaman dalam penerapan Algoritma AES, maka diambil plainteks dan kunci sebagai berikut :

Plainteks = Carles

Kunci = Andi

Penyelesaian :

Tabel 3.3 Plainteks Konversi ke Hexadecimal

Plainteks	Hexadecimal
C	43
a	61
r	72
l	6c
e	65
s	73

Karena jumlah karakter kurang dari 16 maka ditambah dengan 04. Masukkan ke dalam tabel 4x4 sebagai berikut :

43	65	04	04
61	73	04	04

72	04	04	04
6c	04	04	04

Tabel 3.4 Kunci Konversi ke Hexadecimal

Kunci	Hexadecimal
A	41
n	6e
d	64
i	69

Karena jumlah karakter kunci kurang dari 16 karakter maka ditambah dengan 08. Masukkan ke dalam tabel 4x4 sebagai berikut :

41	08	08	08
6e	08	08	08
64	08	08	08
69	08	08	08

Sebelum melakukan enkripsi hitung ekspansi kunci (key schedule) terlebih dahulu seperti :

Key schedule round 1: 9d e6 8a 48 | f8 8b d5 21 | 96 ed ba 53 | fb 8c c9 3a,

Key schedule round 2 : fb 3b 0a 47 | 03 b0 df 66 | 95 5d 65 35 | 6e d1 ac 0f

Key schedule round 3 : c1 aa 7c d8| c2 1a a3 be| 57 47 c6 8b| 39 96 6a 84,

Key schedule round 4 : 58 a8 23 ca | 9a b2 80 74 | cd f5 46 ff | f4 63 2c 7b,

Key schedule round 5 : b3 d9 02 75 | 29 6b 82 01 | e4 9e c4 fe | dd 08 ae 7a,

Key schedule round 6: a3 3d d8 b4 | 8a 56 5a b5 | 6e c8 9e 4b | b3 c0 30 31

Key schedule round 7 : 59 39 1f d9 | d3 6f 45 6c | bd a7 db 27 | 0e 67 eb 16,

Key schedule round 8: 5c d0 58 72 | 8f bf 1d 1e | 32 18 c6 39 | 3c 7f 2d 2f,

Key schedule round 9: 95 08 4d 99 | 1a b7 50 87 | 28 af 96 be | 14 d0 bb 91,

Key schedule round 10: d3 e2 cc 63 | c9 55 9c e4 | e1 fa 0a 72 | f5 2a b1 e3

AddRoundKey atau juga bisa disebut sebagai initial round Plaintext, Kunci :

43 xor 41 = 01000011 xor 01000001= 00000010, hexa: 2

61 xor 6e = 01100001 xor 01101110 = 00001111, hexa: f,

72 xor 64 = 01110010 xor 01100100 = 00010110, hexa: 16

6c xor 69 = 01101100 xor 01101001= 00000101, hexa: 5

65 xor 08 = 01100101 xor 00001000 =01101101, hexa: 6d

73 xor 08 = 01110011 xor 00001000 =01111011, hexa: 7b

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

04 xor 08 = 00000100 xor 00001000 =00001100, hexa: c

Hasil Add Round

2	6d	c	c
f	7b	c	c
16	c	c	c
5	c	c	c

Tahap selanjutnya adalah SubBytes yaitu mengubah hasil AddRoundKey menggunakan tabel S-Box seperti :

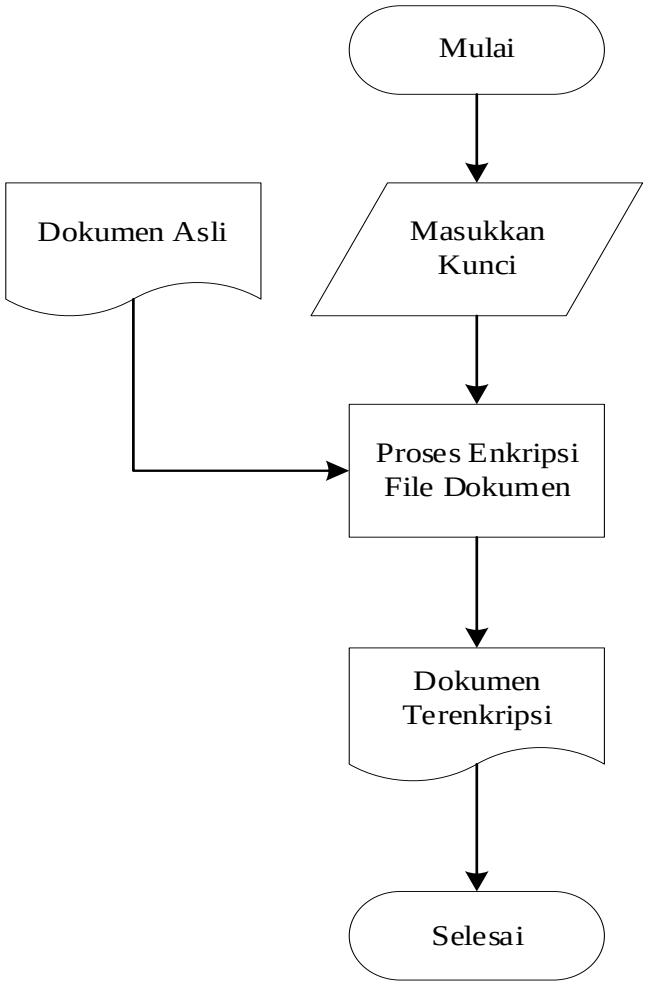
		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Add Round	**Sub Bytes**
2	77

f	76
16	47
5	6b
6d	3c
7b	21
c	ba
c	ba
c	ba
c	ba
c	ba
c	ba
c	ba
c	ba
c	ba
c	ba

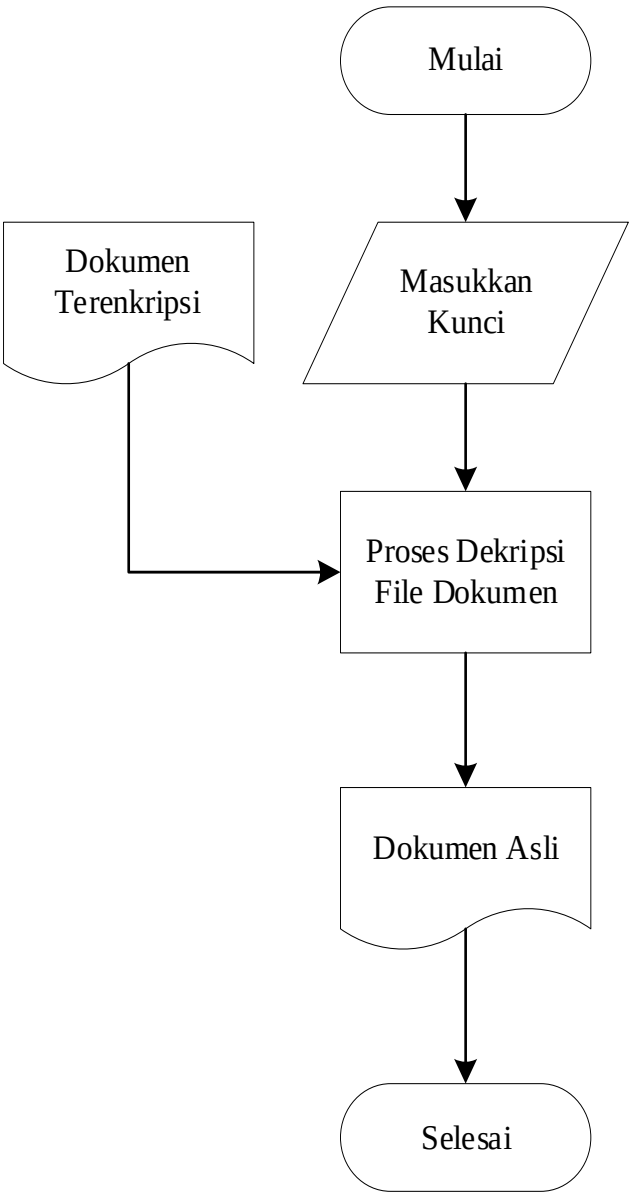
Hasil AddRoundKey() ini adalah state pada ronde 1. Pada AddRoundKey() yang berikut, maka 128 bit yang sudah mengalami perubahan pada ketiga proses tersebut kembali akan di-XOR-kan dengan kunci hasil Ekspand Key kedua dan seterusnya. Dan menghasilkan output chipertext : ’Ývf¯£nüÐ_„Ñ«»'{

C. Flowchart Proses Enkripsi



Gambar 3.3 Flowchart Proses Enkripsi

D. Flowchart Proses Dekripsi

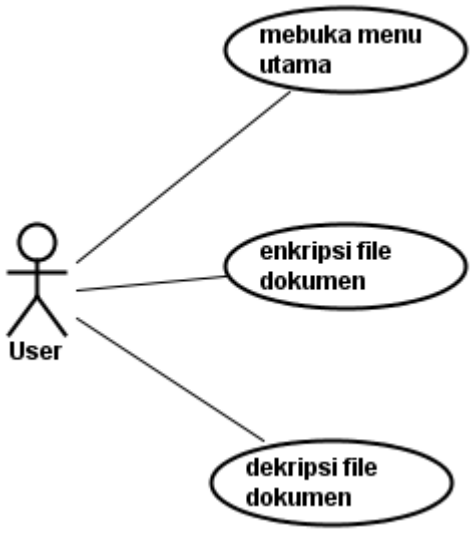


Gambar 3.4. Flowchart Proses Dekripsi

E. Perbandingan Algoritma DES dan AES

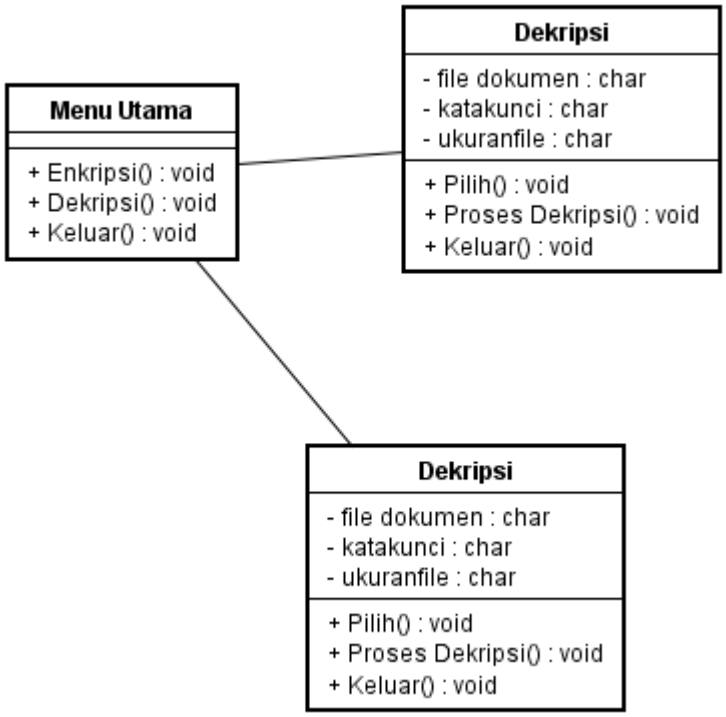
Perbandingan terhadap algoritma DES dan AES menggunakan data pendukung file dokumen dengan ektensi file *.docx, *.ppt, *.pdf, *.wav, *.mp4, *.jpeg, *.png, dimana masing-masing terdiri dari 5 (lima) file. Adapun aspek perbandingan seperti Tabel 3.6.

F. Use Case Diagram



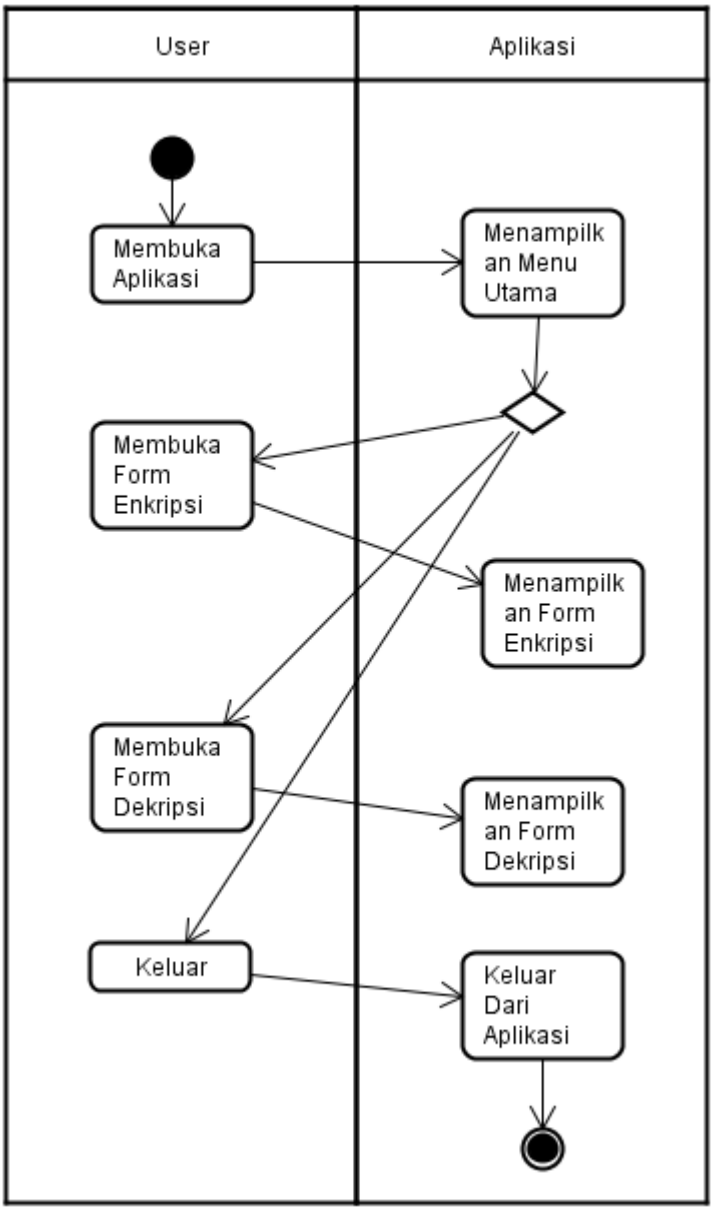
Gambar 3.5. Use Case Diagram

G. Class Diagram



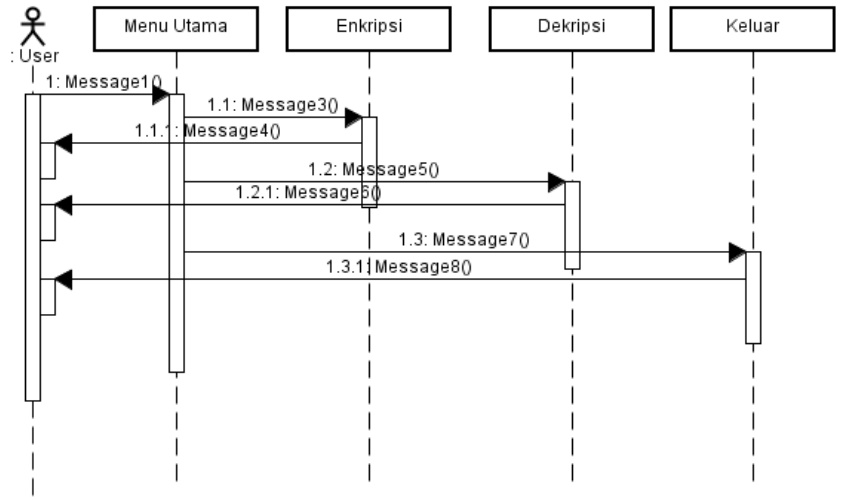
Gambar 3.6. Class Diagram

H. Activity Diagram



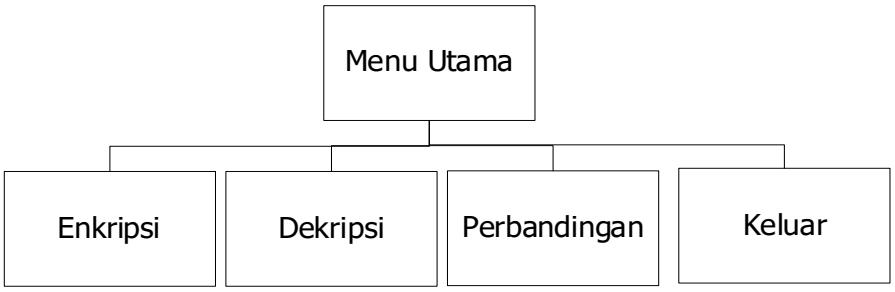
Gambar 3.7. Activity Diagram

I. Sequence Diagram



Gambar 3.8. Sequence Diagram

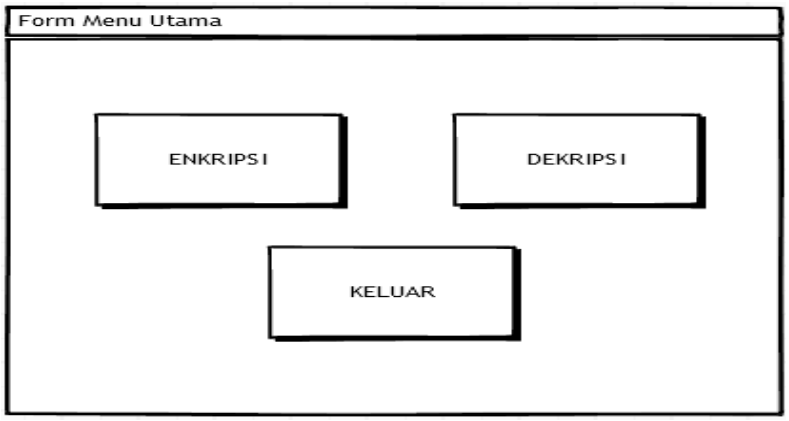
J. Rancangan Struktur Menu



Gambar 3.9. Rancangan Struktur Menu

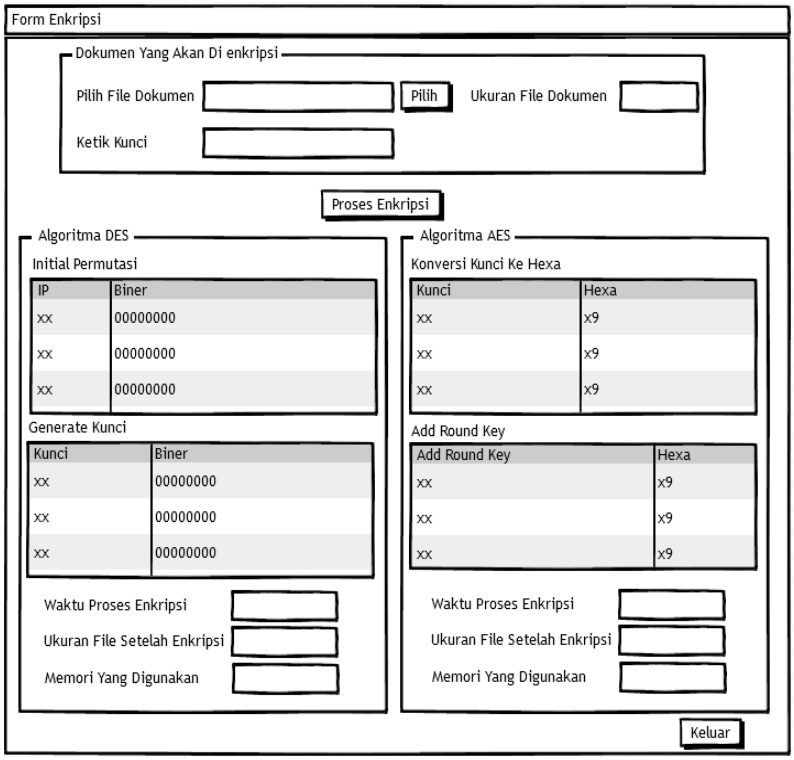
K. Perancangan Aplikasi

1) Menu Utama



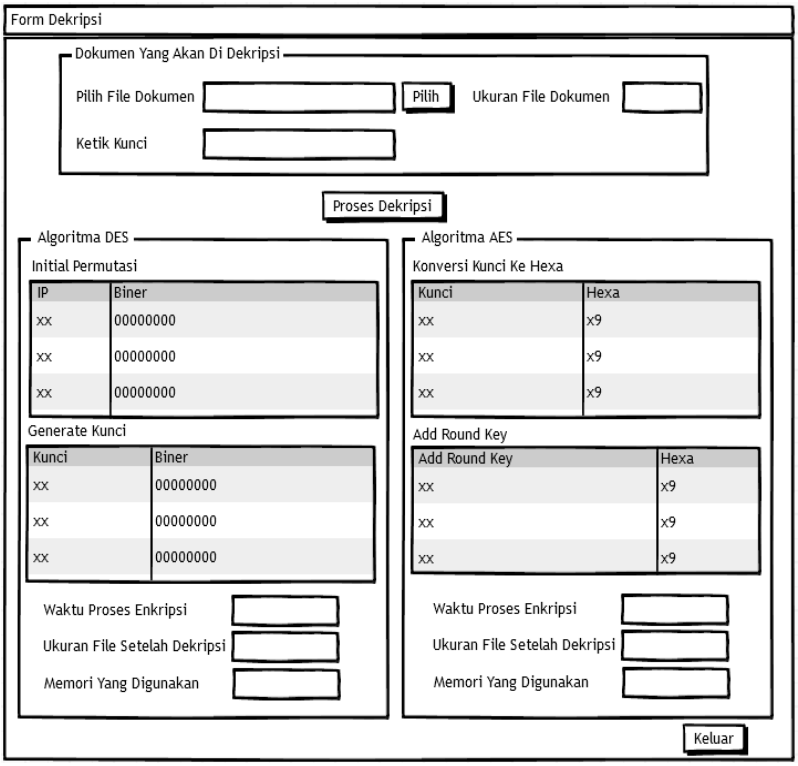
Gambar 3.10. Menu Utama

2) Form Enkripsi



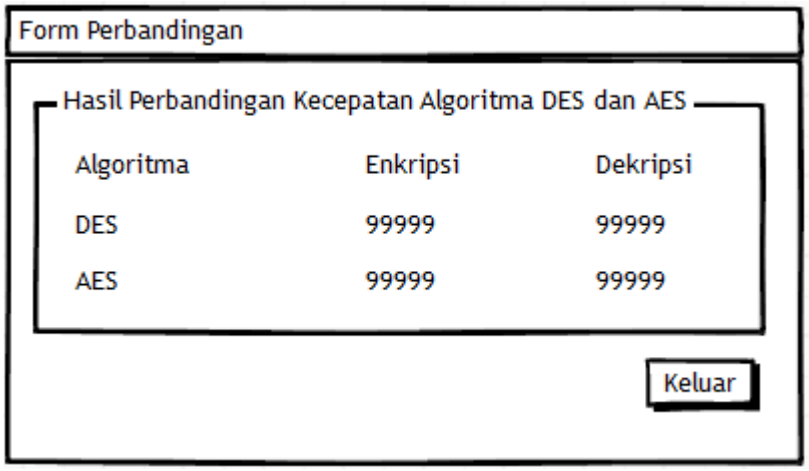
Gambar 3.11. Form Enkripsi

3) Form Dekripsi



Gambar 3.12. Form Dekripsi

4) Form Perbandingan



Gambar 3.13. Form Perbandingan

3.6. Metode Pengujian Sistem

Pengujian sistem merupakan proses eksekusi sistem dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan

mengevaluasi kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau *output* yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Metode pengujian yang dipakai dalam sistem ini adalah metode *black box*. Adapun komponen pengujian yang dilakukan seperti Tabel 3.5.

Tabel 3.5. Pengujian Sistem

No	Komponen Yang Diuji	Skenario Pengujian	Hasil Pengujian
1	Form enkripsi	Enkripsi file dokumen yang akan di enkripsi	
2	Form dekripsi	Dekripsi file dokumen yang telah di enkripsi	
3	Aspek perbandingan waktu	Melakukan uji sebanyak 35 data uji untuk mendapatkan nilai rata-rata waktu proses enkripsi dan dekripsi dari algoritma DES dan AES	
4	Aspek perbandingan ukuran file	Melakukan uji sebanyak 35 data uji untuk mengetahui selisih ukuran file setelah proses enkripsi dan dekripsi dilakukan dari algoritma DES dan AES	
5	Aspek perbandingan memori yang digunakan	Melakukan uji sebanyak 35 data uji untuk mendapatkan nilai rata-rata memori yang digunakan proses enkripsi dan dekripsi dari algoritma	

		DES dan AES	

Tabel 3.6. Analisis Perbandingan Algoritma DES dan AES

Jenis Dokumen	Nama File	Aspek Perbandingan					
		Algoritma DES			Algoritma AES		
		Waktu Proses	Ukuran File	Memori Yang Digunakan	Waktu Proses	Ukuran File	Memori Yang Digunakan
ektensi file *.docx	File 1						
	File 2						
	File 3						
	File 4						
	File 5						
ektensi file *.ppt	File 1						
	File 2						
	File 3						
	File 4						
	File 5						
ektensi file *.pdf	File 1						
	File 2						
	File 3						
	File 4						
	File 5						
ektensi file *.wav	File 1						
	File 2						
	File 3						
	File 4						
	File 5						
ektensi file *.mp4	File 1						
	File 2						
	File 3						

	File 4						
	File 5						
ektensi file *.jpeg	File 1						
	File 2						
	File 3						
	File 4						
	File 5						
ektensi file *.png	File 1						
	File 2						
	File 3						
	File 4						
	File 5						