ANALISIS PERFORMANSI ENKRIPSI PADA PROSESOR INTEL DENGAN ARSITEKTUR X86

INTEL PROCESSOR WITH X86 ARCHITECTURE PERFORMANCE ANALYSIS FOR ENCRYPTION

Shafira Febriani [1], Nyoman Bogi Aditya Karna, S.T., MSEE [2], Ramdhan Nugraha, S.Pd., M.T [3]
[1,2] Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[3] Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
[1]**shfebri@student.telkomuniversity.ac.id**, [2]**aditya@telkomuniversity.co.id**,
[3]**ramdhan@telkomuniversity.ac.id**

Abstrak

Enkripsi adalah sebuah proses yang melakukan perubahan informasi dengan algoritma tertentu sehingga menjadi sebuah kode yang tidak terbaca, hanya orang yang mempunyai kunci penerjemahnya yang bisa membacanya. Proses enkripsi dan dekripsi membutuhkan kemampuan komputasi yang tinggi, hal ini berpengaruh pada kecepatan dan berbagai aspek performansi. Jika diterapkan dalam prosesor, masalah yang diangkat dalam penelitian ini adalah algoritma enkripsi mana yang paling tepat digunakan untuk enkripsi data.

Untuk bisa mengetahui algoritma mana yang lebih efisien, diimplementasikan algoritma *Data Encyption Standard* (DES) dan *Advanced Encryption Standard* (AES) pada aplikasi Cygwin. Aplikasi Cygwin berfungsi untuk meng*compile* program. Pengujian dilakukan dengan membuat program enkripsi untuk AES 128 dan DES dengan bahasa assembly x86. Penelitian ini membandingkan hasil enkripsi untuk parameter komposisi instruksi dan kecepatan komputasi enkripsi pada kedua program.

Hasil pengujian diperoleh algoritma AES membutuhkan lebih sedikit instruksi. Dengan hasil 1537 instruksi pada skenario enkripsi data input dan output terpisah, dan 1487 instruksi pada skenario enkripsi data overwrite. Dengan jenis instruksi yang paling banyak dipakai kedua algoritma adalah Data Transfer. Serta algoritma AES membutuhkan waktu lebih cepat dibandingkan DES. Dengan hasil rata-rata waktu komputasi pada skenario data input dan output terpisah, waktu yang dibutuhkan adalah 0.0544653 detik. Sedangkan pada skenario overwrite data, waktu yang dibutuhkan adalah 0.0520902 detik.

Kata kunci : Enkripsi, AES-128, DES, x86

Abstract

Encryption is a process that changes the information with a certain algorithm so that it becomes a code that is unreadable, only the person who has the key of his interpreter who can read it. The process of encryption and decryption require a high computing capabilities, it does affect the speed and various aspects of performance. If applied in the microprocessor, the issues raised in this study is an encryption algorithm which is most appropriately used for data encryption.

To be able to know which algorithm is more efficient, implemented algorithms Data Encyption Standard (DES) and Advanced Encryption Standard (AES) in Cygwin application. Cygwin applications serve to compile the program. Testing was conducted by creating an encryption program for AES 128 and DES with an x86 assembly language. This project compares the result of the encryption for instruction composition parameters and the computation speed in both programs.

The result obtained AES algorithm requires fewer instructions. With 1537 instructions on the encryption on separate input and output scenario and 1487 instructions on overwrite data scenario. With the most used instruction type in both algorithms are Data Transfer. As well AES compute faster than DES. With the average results on separate input and output scenario the required time is 0.0544653 second.

Keywords: Encryption, AES-128, DES, x86

1. Pendahuluan

Pada perkembangan *internet of things* (IoT), device dengan arsitektur mikroprosesor Intel x86 sangat cocok digunakan pada *edge computing gateway* dalam infrastruktur untuk menganalis dan menyimpan data penting. Oleh karena itu, dibutuhkan algoritma enkripsi yang efisien dan tidak terlalu memakan memori.

Saat ini, teknologi enkripsi sudah banyak diaplikasikan untuk kepentingan umum, seperti merahasiakan data-data penting milik perorangan maupun perusahaan agar tidak mudah disadap. Enkripsi adalah sebuah proses yang melakukan perubahan informasi dengan algoritma tertentu sehingga menjadi sebuah kode yang tidak terbaca, hanya orang yang mempunyai kunci penerjemahnya yang bisa membacanya. Data asli yang dikirim (belum

dienkripsi) disebut *plaintext*, kemudian data tersebut dienkripsi menggunakan algoritma enkripsi dan kunci enkripsi. Proses tersebut menghasilkan data baru yang disebut *chipertext*.

Algoritma enkripsi mempunyai kelemahan dan kelebihan dalam performansi enkripsi data, maka diperlukan analisis untuk mengukur kinerja program melalui uji performansi sistem dalam proses enkripsi. Dimana algoritma yang efisien ialah algoritma yang meminimumkan kebutuhan ruang penyimpanan dan waktu. Algoritma enkripsi simetris hanya memerlukan satu kunci yang sama pada pengirim dan penerima. Dalam algoritma simetris terdapat beberapa algoritma diantaranya *Advanced Encryption Standard* (AES) dan *Data Encryption Standard* (DES) yang digunakan dalam penelitian Tugas Akhir ini.

Pada penelitian sebelumnya [8], diaplikasikan enkripsi teks dengan AES dan DES pada system operasi Windows dan Ubuntu dengan bahasa pemrograman Java yang diimplementasikan untuk *Smart City*. Dengan parameter uji kecepetan proses enkripsi, diperoleh hasil enkripsi menggunakan system operasi Ubuntu dan algoritma AES bekerja dengan waktu lebih cepat. Penelitian Tugas Akhir ini bertujuan untuk mengembangkan penilitian [8] pada prosesor Intel x86 dengan bahasa Assembly x86, parameter yang diuji adalah membandingkan hasil performansi komposisi instruksi enkripsi dari algoritma DES dengan algoritma AES-128bit pada x86.

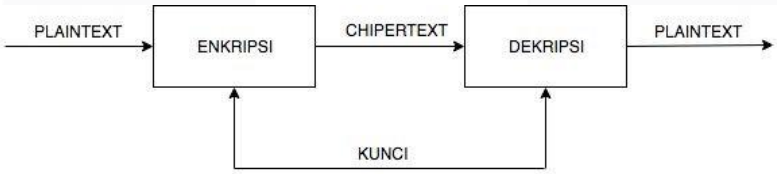
2. Konsep Dasar

2.1 Kriptografi

Kriptografi adalah ilmu mengenai bagaimana cara menjaga kerahasiaan pesan [2]. Dalam kriptografi terdapat enkripsi dan dekripsi. Enkripsi adalah proses data diubah menggunakan kunci enkripsi menjadi data yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi [3]. Sedangkan, dekripsi kebalikan dari proses enkripsi yaitu merubah data yang sudah dienkripsi menjadi data asli, sehingga dapat dibaca kembali. Data asli yang dikirim (belum dienkripsi) disebut *plaintext*, kemudian data tersebut dienkripsi menggunakan algoritma enkripsi dan kunci enkripsi. Proses tersebut menghasilkan data baru yang disebut *ciphertext*. Sehingga enkripsi sangat penting untuk melindungi informasi agar tidak mudah terlihat ataupun disadap pihak yang tidak berhak.

2.2 Algoritma Kriptografi Simetris

Algoritma kriptografi simetris adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi. Algoritma kriptografi simetris dibagi menjadi dua yaitu *stream ciphers* (algoritma aliran) dan *block ciphers* (algoritma blok). Proses penyandian pada *stream ciphers* berorientasi pada satu bit/byte data. Sedangkan pada *block ciphers* berorientasi pada sekumpulan bit/byte data (per blok) [6]. Contoh algoritma simetris adalah *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES) atau Rijndael.



Gambar 2.1 Proses Enkripsi Algoritma Kriptografi Simetris

2.2.1 *Data Encryption Standard*

Data Encryption Standard (DES) adalah suatu blok *cipher* yang membutuhkan serangkaian panjang dan mengubah bit *plaintext* melalui serangkaian operasi ke *bitstring ciphertext* lain yang sama panjang. DES merupakan kombinasi dari dua model enkripsi dasar yaitu substitusi dan permutasi.
Dimana *f* adalah fungsi *cipher*, L_{n-1} adalah blok yang sedang tidak dienkripsi, $\oplus$ adalah operasi XOR, R_{n-1} adalah blok yang sedang dienkripsi, dan K_n adalah kunci untuk putaran *n*.

DES menggunakan kunci sebesar 64-bit, namun hanya 56 diantaranya yang benar-benar digunakan oleh algoritma dan sisa 8-bit digunakan sebagai *parity* [3]. *Parity* digunakan untuk memeriksa apakah kunci tersebut benar (kunci hasil putaran atau kunci *plaintext*). Maka panjang kunci yang efektif adalah 56-bit. Penomoran bit adalah dari kiri ke kanan dengan bit 1 menjadi *most significant bit*. Untuk 64-bit, bit 1 mempunya nilai 2^{63} [5]. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP). Hasil permutasi awal kemudian dibagi menjadi dua blok, yaitu blok L0 dan R0, L0 adalah 32-bit pertama dari hasil permutasi dan R0 adalah 32-bit sisanya. Enkripsi dilakukan sebanyak 16 putaran, setiap putaran menggunakan kunci internal yang berbeda. Hasilnya kemudian dipermutasi dengan matriks *invers initial permutation* atau IP^{-1} menjadi blok cipherteks.

2.2.2 *Advanced Encryption Standard*

Data Encryption Standard (DES) dianggap sudah tidak aman lagi, karena dengan perangkat keras khusus kuncinya bisa ditemukan dalam waktu yang singkat. *Advanced Encryption Standard* (AES) ada dengan harapan dapat lebih aman dan lebih cepat. Perbedaan teknik enkripsi AES dan DES adalah AES menggunakan substitusi (S-Box) secara langsung, sedangkan S-Sox dalam DES hanya dalam fungsi *cipher f* yang kemudian hasilnya

menggunakan XOR. Algoritma AES yang beroperasi pada blok 128-bit dengan kunci 128-bit [4] adalah sebagai berikut, diluar proses pembangkitan *round key* [3]:

1. AddRoundKey: melakukan XOR antara state awal (*plaintext*) dengan *cipher key*. Tahap ini disebut juga *initial round*
2. Putaran sebanyak *Nr–1* kali. Proses yang dilakukan pada setiap putaran, yaitu:
   a. SubBytes: substitusi byte dengan menggunakan tabel substitusi (S-box)
   b. ShiftRows: pergeseran baris-baris *array state* secara *wrapping*.
   c. MixColumns: mengacak data di masing-masing kolom *array state*
   d. AddRoundKey: melakukan XOR antara state sekarang *round key*
3. *Final round* atau proses untuk putaran terakhir, yaitu:
   a. SubBytes
   b. ShiftRows
   c. AddRoundKey

2.3 Arsitektur Mikroprosesor Intel x86

Arsitektur x86 atau 80X86 adalah nama umum dari arsitektur mikroprosesor yang pertama kali dikembangkan dan diproduksi oleh Intel. Arsitektur ini terkenal dengan nama x86, dikarenakan prosesor awal dari keluarga arsitektur ini mempunyai nomor model yang berakhiran angka "86". Arsitektur x86 lahir melalui 8086 CPU pada tahun 1978. Intel 8086 adalah pengembangan dari mikroprosesor Intel 8080.

Arsitektur x86 adalah rancangan *Complex Instruction Set Computer* (CISC) dengan ukuran panjang instruksi yang bervariasi. Arsitektur mikkroprosesor x86 dapat bekerja dalam berbagai kondisi, seperti:

a. *Real mode*, kondisi dimana prosesor x86 bekerja seolah sebuah prosesor Intel 8086 atau 8088, meskipun sebenarnya merupakan prosesor yang lebih tinggi
b. *Protected mode*, kondisi dimana terdapat proteksi untuk alamat memori untuk digunakan oleh sistem operasi
c. *Virtual protected mode*, kondisi ini memungkinkan aplikasi 16-bit untu dapat tetap berjalan pada system operasi.

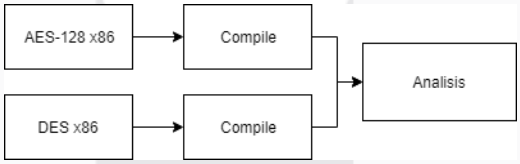
2.4 Cygwin

Cygwin memungkinkan program dengan system mirip Unix untuk dikompilasi ulang dan dijalankan secara native di Windows, dengan modifikasi *source code* dengan menyediakan POSIX API yang sama seperti yang diharapkan dalan sistem. Direktori instalasi Cygwin seperti root dan mengikuti tata letak direktori yang mirip seperti dalam system Unix. Cygwin menyediakan integrase apikasi berbasis Windows, data, dan sumber daya system lainnya dengan aplikasi.

3. Pembahasan

3.1 Desain Sistem

Desain sistem yang akan membahas mengenai enkripsi dengan algoritma DES dan AES menggunakan bahasa Assembly x86. Sistem ini dirancang untuk mengetahui algoritma mana yang efektif untuk enkripsi data berdasarkan instruksi bahasa mesin.

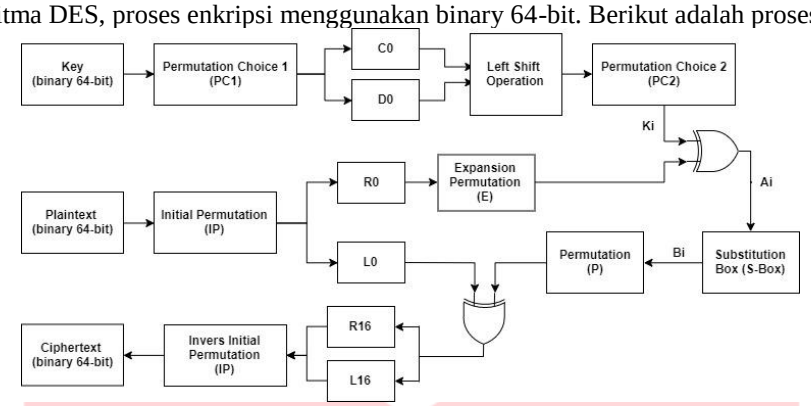


Gambar 3.1 Diagram Blok Sistem

Tahap pertama diawali dengan pembuatan program untuk enkripsi DES dan AES dengan bahasa Assembly x86. Tahap kedua adalah compile program, tahap ini merupakan proses merubah bahasa program (*source code*) menjadi bahasa komputer (*binary*). Yang terakhir adalah tahap ketiga yaitu analisis, tahap ini merupakan analisa hasil dari penelitian sistem.

3.2 Sistem Enkripsi DES

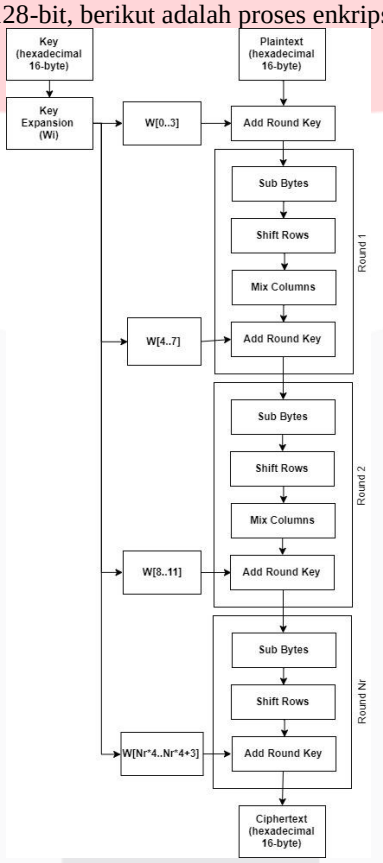
Pada algoritma DES, proses enkripsi menggunakan binary 64-bit. Berikut adalah proses enkripsi DES:



Gambar 3.2 Diagram Blok Enkripsi DES

3.3 Sistem Enkripsi AES

Algoritma AES menggunakan 128-bit, berikut adalah proses enkripsi pada algoritma AES:



Gambar 3.3 Diagram Blok Enkripsi AES-128bit

4. Pengujian dan Analisis Sistem

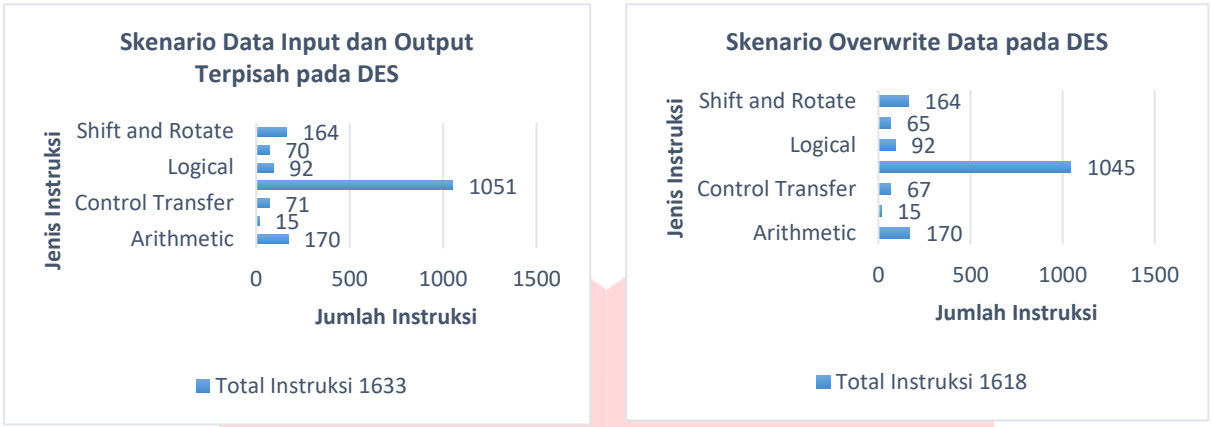
4.1 Assembly x86 pada Cygwin

Pengujian enkripsi *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES) pada Tugas Akhir ini disimulasikan pada terminal Cygwin 64-bit dengan Bahasa Assembly x86. Terminal Cygwin berfungsi untuk *compile* program enkripsi. Pesan yang akan dienkripsi berupa file teks dengan format .txt

Instruksi Bahasa Assembly x86 mempunyai 167 jenis instruksi. Tujuan dari pengujian ini adalah untuk menentukan instruksi spesifik yang digunakan pada enkripsi dengan algoritma *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES), agar tidak ada instruksi yang terbuang.

4.2 Pengujian Enkripsi *Data Encryption Standard* (DES)

Pengujian dilakukan berdasarkan parameter komposisi intruksi aritmatika dan logika pada enkripsi dengan algoritma *Data Encryption Standard* (DES). Pengujian ini bertujuan untuk mengetahui komposisi instruksi yang digunakan untuk enkripsi. Pengujian dibagi menjadi dua scenario yaitu data input dan output terpisah dan scenario overwrite data.

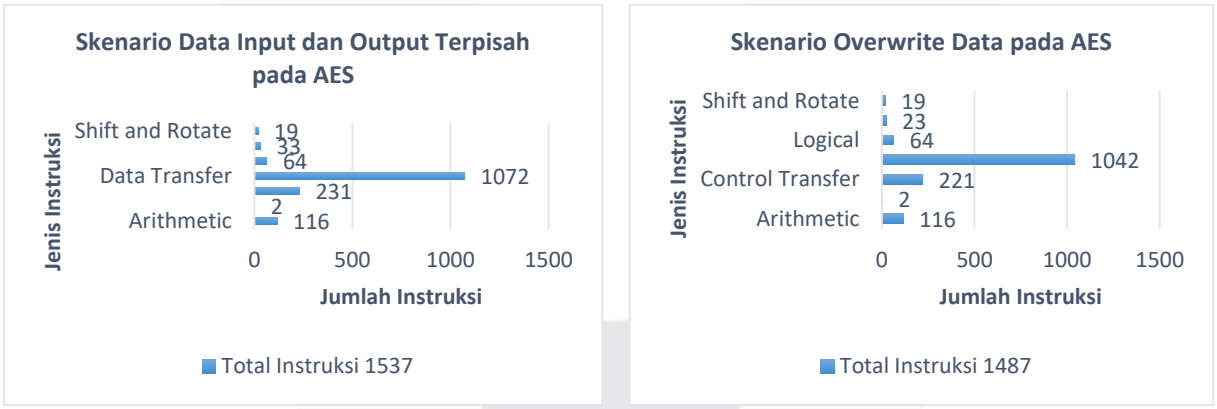


Gambar 4.1 Hasil Skenario pada DES

Berdasarkan hasil pengujian pada skenario data input dan ouput terpisah, diperoleh total instruksi yang digunakan sebanyak 1633 instruksi. Sedangkan pada scenario overwrite data, total instruksi yang digunakan sebanyak 1618 instruksi.

4.3 Pengujian Enkripsi *Advanced Encryption Standard* (AES)

Pengujian dilakukan berdasarkan parameter komposisi intruksi aritmatika dan logika pada enkripsi dengan algoritma *Advanced Encryption Standard* (AES). Pengujian ini bertujuan untuk mengetahui komposisi instruksi yang digunakan untuk enkripsi. Pengujian dibagi menjadi dua skenario yaitu data input dan output terpisah dan skenario overwrite data.



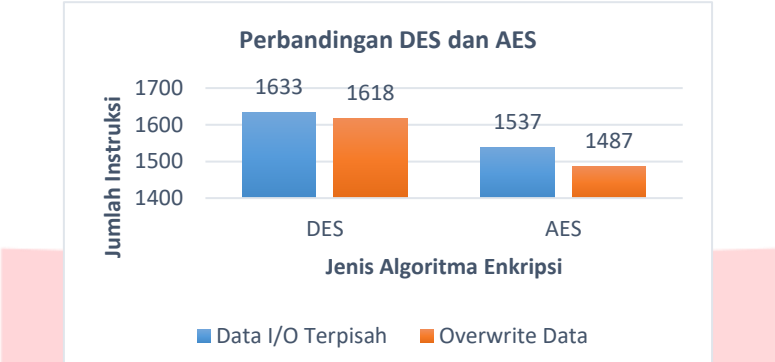
Gambar 4.2 Hasil Skenario pada AES

Berdasarkan hasil pengujian pada skenario data input dan output terpisah, diperoleh total instruksi yang digunakan sebanyak 1537 instruksi. Sedangkan pada skenario overwrite data, total instruksi yang digunakan sebanyak 1487 instruksi.

4.4 Perbandingan Pengujian Parameter Komposisi Instruksi

Pengujian parameter komposisi instruksi enkripsi pada algoritma *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES) dengan dua skenario yaitu data input dan output terpisah dan data overwrite.

Berdasarkan hasil pengujian, diperoleh hasil algoritma AES membutuhkan lebih sedikit instruksi dibandingkan algoritma DES. Dengan hasil 1537 instruksi pada skenario enkripsi data input dan output terpisah, dan 1487 instruksi pada skenario enkripsi data overwrite. Jenis instruksi yang paling banyak digunakan pada kedua algoritma adalah Data Transfer.



Gambar 4.3 Hasil Pengujian Komposisi Instruksi

4.5 Pengujian Parameter Kecepatan

Pengujian dilakukan berdasarkan parameter kecepatan komputasi pada program enkripsi dengan algoritma DES dan AES. Pengujian ini bertujuan untuk mengetahui kecepatan komputasi yang digunakan untuk enkripsi sehingga didapatkan program yang efisien dari segi waktu. Pada masing-masing skenario, pengujian dilakukan sebanyak 10 kali lalu hasil yang diperoleh dirata-ratakan.

Berdasarkan hasil pengujian kecepatan diperoleh hasil bahwa algoritma AES membutuhkan waktu lebih cepat dibandingkan DES. Dengan hasil rata-rata waktu komputasi pada skenario data input dan output terpisah, waktu yang dibutuhkan adalah 0.0544653 detik. Sedangkan pada skenario overwrite data, waktu yang dibutuhkan adalah 0.0520902 detik.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Parameter pengujian yang dilakukan adalah komposisi instruksi dan kecepatan komputasi program pada enkripsi algoritma *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES). Pengujian dilakukan dengan dua skenario enkripsi yaitu data input dan output terpisah dan data overwrite.

Jenis instruksi yang digunakan dibagi menjadi 7 jenis, yaitu Arithmetic, Bit and Byte, Control Transfer, Data Transfer, Logical, Miscellanous, Shift and Rotate. Algoritma AES membutuhkan lebih sedikit instruksi dibandingkan algoritma DES. Dengan hasil 1537 instruksi pada skenario enkripsi data input dan output terpisah, dan 1487 instruksi pada skenario enkripsi data overwrite. Jenis instruksi yang paling banyak digunakan pada kedua algoritma adalah Data Transfer.

Algoritma AES membutuhkan waktu lebih cepat dibandingkan DES. Dengan hasil rata-rata waktu komputasi pada skenario data input dan output terpisah, waktu yang dibutuhkan adalah 0.0544653 detik. Sedangkan pada skenario overwrite data, waktu yang dibutuhkan adalah 0.0520902 detik. Sehingga algoritma AES lebih efisien dari segi komposisi instruksi dan kecepatan komputasi enkripsi.

5.2 Saran

Adapun saran untuk pengembangan Tugas Akhir ini dan pengembangan penelitian selanjutnya, yaitu:

1. Format file enkripsi selanjutnya dapat berupa format file jenis lain
2. Penelitian selanjutnya dapat melakukan uji performansi daya yang digunakan
3. Dapat diterapkan pada produk IoT

Daftar Pustaka:

[1] R. W. Wardhani, D. Ogi, M. Syahral, and P. Dedy Septono Catur, "Fast implementation of AES on Cortex-M3 for security information devices," *QiR 2017 - 2017 15th Int. Conf. Qual. Res. Int. Symp. Electr. Comput. Eng.*, vol. 2017–Decem, pp. 241–244, 2017.
[2] B. Bhat, A. W. Ali, and A. Gupta, "DES and AES performance evaluation," *Int. Conf. Comput. Commun. Autom. ICCCA 2015*, pp. 887–890, 2015.
[3] Y. Zhang, "Design and Implementation of AES based on ARM920T Processor," no. 2, pp. 3–7, 2015.
[4] B. K. B. Raju, A. Krishna, and G. Mishra, "Implementation of an efficient dynamic AES algorithm using ARM based SoC," *2017 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017*, vol. 2018–Janua, pp. 39–43, 2018.
[5] S. Kromodimoejo, "Teori & Aplikasi Kriptografi". Januari 2010.
[6] B. Schneier, "Applied Cryptography".
[7] J. Yiu, "The Definitive Guide to the ARM Cortex-M3 2nd Edition", Elseiver Inc, 2010.
[8] Dadhich, S. (2016). *Performance Analysis of AES and DES Cryptographic Algorithms on Windows & Ubuntu using Java*. *35*(4), 179–183.
[9] ORACLE, "x86 Assembly Language Reference Manual", 2010.
[10] Intel, "Intel 64 and IA-32 Architectures Software Developer's Manual", September 2016.
[11] Intel, "Appendix A: Intel x86 Instruction Reference"