Pengaruh Variasi Panjang Kunci, Ukuran Blok, dan Mode Operasi Terhadap Waktu Eksekusi pada Algoritma Rijndael

Trihastuti Yuniati
Jurusan Informatika
Fakultas MIPA, UNS
Jl. Ir. Sutami 36 A Kentingan
Surakarta
three.sakha@gmail.com

Esti Suryani
Jurusan Informatika
Fakultas MIPA, UNS
Jl.Ir. Sutami 36 A Kentingan
Surakarta
suryapalapa@yahoo.com

Abdul Aziz
Jurusan Informatika
Fakultas MIPA, UNS
Jl. Ir. Sutami 36 A Kentingan
Surakarta
Abdul_7773@yahoo.com

ABSTRAK

Algoritma Rijndael merupakan salah satu algoritma kriptografi yang berjalan pada mode operasi cipher blok. Rijndael mendukung panjang kunci dan ukuran blok 128-bit sampai 256-bit dengan step 32 bit. Paper ini membahas bagaimana pengaruh variasi panjang kunci, ukuran blok dan mode operasi terhadap waktu eksekusi pada algoritma Rijndael. Eksperimen dilakukan terhadap empat berkas pdf berukuran berbeda, 2.5 MB, 5 MB, 10 MB, dan 20 MB. Keempat berkas tersebut dilakukan enkripsi dan dekripsi dengan berbagai kombinasi panjang kunci, ukuran blok, dan mode operasi. Variasi panjang kunci dan ukuran blok adalah 128-bit, 192-bit, dan 256-bit, dan variasi mode operasi adalah ECB, CBC, dan CFB. Tiap kombinasi diulang lima kali untuk mendapatkan waktu eksekusi rata-ratanya. Hasil penelitian menunjukkan bahwa kecepatan eksekusi pada mode ECB dan CBC sangat dipengaruhi oleh jumlah putaran, dimana jumlah putaran tergantung pada panjang kunci dan ukuran blok, sedangkan kecepatan eksekusi pada mode CFB relatif dipengaruhi oleh ukuran blok.

Keywords

algoritma Rijndael, cipher blok, kriptografi, mode operasi

1. PENDAHULUAN

Data merupakan salah satu aset yang sangat penting bagi siapapun, baik itu perusahaan, instansi pemerintahan, maupun institusi pendidikan. Data ini ada yang bersifat terbuka, artinya boleh diketahui oleh semua orang, dan ada yang bersifat rahasia, dimana hanya orang-orang tertentu yang boleh mengetahuinya. Data yang bersifat rahasia ini membutuhkan metode khusus untuk menjaga kerahasiaannya.

Berbagai upaya pengamanan data telah dilakukan untuk menjaga kerahasiaan data. Salah satu cara yang digunakan adalah dengan menyandikan data menjadi kode-kode yang tidak dimengerti. Penyandian dilakukan agar apabila data tersebut jatuh ke tangan pihak yang tidak berhak, pihak tersebut tetap tidak dapat memahami informasi yang sesungguhnya. Metode pengamanan data ini dikenal sebagai metode kriptografi. Berbagai macam algoritma kriptografi telah dikembangkan. Algoritma kriptografi yang baik akan memerlukan waktu yang lama untuk memecahkan data yang telah disandikan. Seiring dengan perkembangan teknologi komputer, dunia teknologi informasi membutuhkan algoritma kriptografi yang lebih kuat dan lebih aman, salah satunya adalah Rijndael. Algoritma ini memiliki keunggulan dalam hal performansi dan kesederhanaan kode, serta tingkat keamanan data yang tinggi untuk ukuran teknologi komputer yang ada saat ini [1], [2].

Sebagaimana algoritma blok cipher pada umumnya, algoritma Rijndael dapat dijalankan dalam beberapa mode operasi [3], [4]. Menurut [4] menunjukkan bahwa algoritma Rijndael dapat dijalankan dalam empat mode operasi, yaitu *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB), kunci 128-*bit*, 192-*bit*, dan 256-*bit*, dengan blok 128-*bit*. Menurut [5] dan [6] menunjukkan bahwa algoritma Rijndael mendukung variasi panjang kunci dan ukuran blok dari 128-*bit* sampai 256-*bit* dengan step 32-*bit*. Panjang kunci dan ukuran blok mempengaruhi jumlah putaran dalam proses enkripsi maupun dekripsi.

Analisa dilakukan terhadap algoritma Rijndael dengan ukuran blok 128-*bit* dan kunci 128-*bit*, 192-*bit*, dan 256-*bit* dijalankan dalam empat mode operasi (ECB, CBC, CFB, dan OFB) [5], sedangkan di dalam [6], analisa dilakukan terhadap algoritma Rijndael yang mendukung panjang kunci dan ukuran blok 128-*bit*, 192-*bit*, dan 256-*bit*, namun tidak ada variasi mode operasi. Kedua penelitian tersebut belum meneliti bagaimana pengaruh kombinasi dari variasi ketiga variabel (panjang kunci, ukuran blok, dan mode operasi) terhadap waktu eksekusi. Berangkat dari hal tersebut, maka dalam penelitian ini penulis bermaksud untuk menganalisa bagaimanakah pengaruh variasi panjang kunci, ukuran blok, dan mode operasi yang digunakan terhadap waktu eksekusi pada algoritma Rijndael.

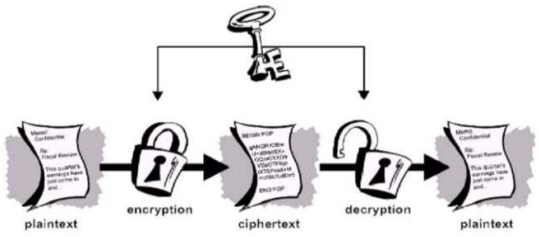
2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi secara umum adalah ilmu dan seni untuk menjaga keamanan pesan [7]. Sedangkan definisi kriptografi menurut [8], kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data.

Berdasarkan kunci yang dipakai, algoritma kriptografi dibagi menjadi dua [7], yaitu:

1. Algoritma kunci simetris
   Algoritma ini sering disebut sebagai algoritma klasik karena kunci yang digunakan untuk enkripsi sama dengan kunci untuk dekripsi. Keamanan dari pesan sangat tergantung pada kuncinya. Proses enkripsi-dekripsi dengan algoritma kunci simetris ditunjukkan oleh Gambar 1.



Gambar 1 Proses enkripsi-dekripsi dengan algoritma kunci simetris [15]

Algoritma kriptografi simetri yang beroperasi pada mode *bit* dapat dikelompokkan dalam dua kategori, yaitu:

a. Cipher aliran (*stream cipher*)
   Algoritma kriptografi yang beroperasi dalam bentuk *bit* tunggal. Rangkaian *bit* dienkripsikan *bit* per *bit*.
b. Cipher blok (*block cipher*)
   Algoritma kriptografi yang beroperasi dalam bentuk blok *bit*. Rangkaian *bit* dibagi menjadi blok-blok *bit* yang panjangnya sudah ditentukan sebelumnya.
2. Algoritma kunci asimetris
   Pada algoritma kunci asimetris, kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Pada algoritma ini, terdapat dua macam kunci:
   a. Kunci publik (*public key*), yaitu kunci yang dipublikasikan dan boleh diketahui oleh semua orang.
   b. Kunci pribadi (*private key*), yaitu kunci yang dirahasiakan dan hanya diketahui oleh satu orang.

2.2 *Cipher* Blok

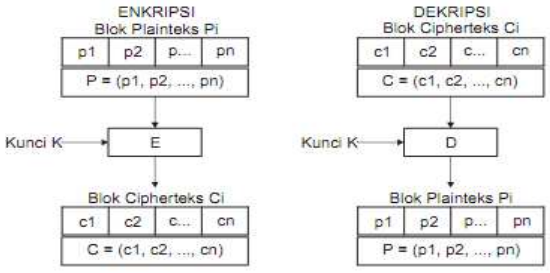
Penyandian blok pada dasarnya adalah proses penyandian terhadap blok-blok data yang jumlahnya sudah ditentukan. Pada awalnya (sekitar tahun 1980), terdapat empat mode operasi dalam sistem penyandian blok, yaitu *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB). Mode ini distandarisasi oleh FIPS 81, ANSI X3.106, dan ISO/IEC 10116. Kemudian muncul dua mode baru, yaitu *Counter Mode* (CTR) dan *Cipher Text Stealing* (CTS). CTR dan CTS distandarisasi dalam NIST SP800-38A (SP800-38A mengakui keempat mode dari FIPS 81, ketika ISO 10116 diperbarui). Mode operasi menentukan bagaimana keluaran dari putaran sebelumnya digunakan sebagai masukan bagi putaran selanjutnya [7], [9]

2.3 Electronic Code Block (ECB)

Mode ECB adalah mode yang paling umum dan paling mudah untuk diimplementasikan. Plainteks dibagi ke dalam blok-blok yang ukurannya telah ditentukan. Setiap blok *plainteks* P_i dienkripsi secara individual dan independen menjadi blok *cipherteks* C_i menggunakan kunci yang diberikan. Secara matematis proses enkripsi-dekripsi pada mode ECB dituliskan seperti persamaan 2.6 dan 2.7 berikut:

Enkripsi: $C_i = E_k(P_i)$
Dekripsi: $P_i = D_k(C_i)$

yang dalam hal ini, P_i dan C_i masing-masing adalah blok *plainteks* dan *cipherteks* ke-*i*. Skema enkripsi dan dekripsi dengan mode ECB ditunjukkan oleh Gambar 3.



Gambar 2 Skema enkripsi dan dekripsi dengan mode ECB

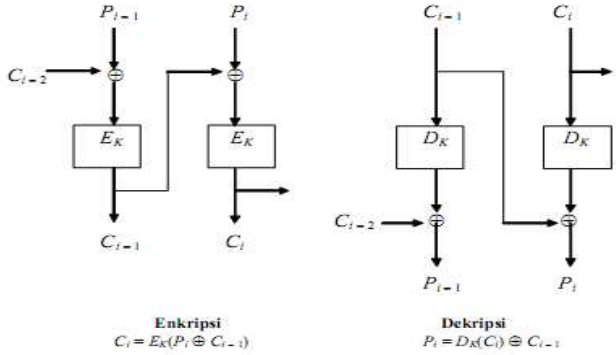
2.4 Cipher Block Chaining (CBC)

Pada CBC digunakan operasi umpan balik atau dikenal dengan operasi berantai (*chaining*). Tujuan dari mode ini adalah untuk membuat ketergantungan antarblok. Setiap blok *cipherteks* tidak hanya bergantung pada blok *plainteks*-nya tetapi juga pada seluruh blok *plainteks* sebelumnya. Hasil enkripsi blok sebelumnya diumpanbalikkan ke dalam enkripsi blok saat ini (*current*). Caranya, blok *plainteks current* di-*XOR*-kan terlebih dahulu dengan blok *cipherteks* hasil enkripsi sebelumnya, selanjutnya hasil peng-*XOR*-an ini masuk ke dalam fungsi enkripsi. Secara matematis enkripsi-dekripsi dengan mode CBC dinyatakan seperti pada persamaan berikut:

Enkripsi: $C_i = E_k(P_i \oplus C_{i-1})$

Dekripsi: $P_i = D_k(C_i) \oplus C_{i-1}$

yang dalam hal ini $C_0 = IV$ (*initialization vector*). Skema enkripsi dan dekripsi dengan mode CBC ditunjukkan oleh Gambar 3.



Gambar 3.Skema enkripsi dan dekripsi dengan mode CBC [10]

2.5 *Cipher Feedback* (CFB)

Mode CFB mengatasi kelemahan pada mode CBC jika diterapkan pada komunikasi data. Pada mode CBC, proses enkripsi atau dekripsi tidak dapat dilakukan sebelum blok data yang diterima lengkap terlebih dahulu. Masalah ini diatasi pada mode CFB. Pada mode ini, data dapat dienkripsi pada unit-unit yang lebih kecil atau sama dengan ukuran satu blok. Unit yang dienkripsikan dapat berupa *bit* per *bit* (seperti *stream cipher*). Misalkan pada CFB 8-*bit*, maka data akan diproses tiap 8-*bit*. Secara matematis proses enkripsi-dekripsi mode CFB dinyatakan seperti pada persamaan berikut:

Enkripsi (E_k):

$$C_i = P_i \oplus MSB_m(E_k(X_i))$$
$$X_{i+1} = LSB_{m-n}(X_i)||C_i$$

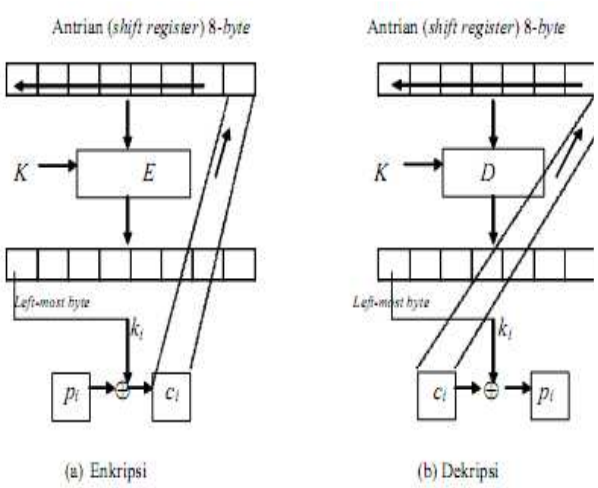
Dekripsi (D_k):

$$P_i = C_i \oplus MSB_m(D_k(X_i))$$
$$X_{i+1} = LSB_{m-n}(X_i)||C_i$$

yang dalam hal ini:

X_i = isi antrian dengan X_1 adalah *IV*
E = fungsi enkripsi dengan algoritma cipher blok
D = fungsi dekripsi dengan algoritma cipher blok
K = kunci
m = panjang blok enkripsi/dekripsi
n = panjang unit enkripsi/dekripsi
|| = operator penyambungan (*concatenation*)
MSB = *Most Significant Byte*
LSB = *Least Significant Byte*

Mode CFB membutuhkan sebuah antrian (*queue*) yang berukuran sama dengan ukuran blok masukan. Skema enkripsi dan dekripsi dengan mode CFB 8-*bit* yang bekerja pada blok berukuran 64-*bit* (setara dengan 8-*byte*) ditunjukkan oleh Gambar 4.



Gambar 4: Skema enkripsi dan dekripsi dengan mode CFB [13]

2.6 Kotak Substitusi (*S-Box*)

S-Box merupakan matriks yang berisi substitusi sederhana yang memetakan satu atau lebih *bit* dengan satu atau lebih *bit* yang lain [10]. Pada mayoritas algoritma cipher blok, *S-Box* memetakan m-*bit* masukan menjadi n-*bit* keluaran, sehingga dinamakan sebagai *m x n S-Box*. Operasi *S-Box* adalah dengan *look-up table*, sehingga *S-Box* menjadi satu-satunya langkah non-linear di dalam algoritma. *S-Box* harus dirancang dengan baik sedemikian sehingga memiliki kekuatan kriptografi yang bagus dan mudah diimplementasikan. Ada empat cara yang dapat digunakan untuk merancang *S-Box*, yaitu:

1. Dipilih secara acak, cara ini cukup aman untuk *S-Box* yang berukuran besar, namun tidak aman untuk *S-Box* berukuran kecil.
2. Dipilih secara acak kemudian diuji, cara ini sama dengan cara pertama, hanya saja nilai acak yang dibangkitkan akan diuji apakah memenuhi sifat tertentu.
3. Dibuat oleh orang (*man-made*), dimana masukan di dalam *S-Box* dibangkitkan dengan teknik yang lebih intuitif.
4. Dihitung secara matematis (*math-mode*), dimana masukan di dalam *S-Box* dibangkitkan berdasarkan prinsip matematika.

2.7 Algoritma Rijndael

Algoritma Rijndael (**dibaca: Rhine-doll**) merupakan algoritma yang didesain oleh Vincent Rijmen dan John Daemen asal Belgia dalam rangka mengikuti kompetisi algoritma kriptografi pengganti DES yang diadakan oleh NIST (*National Institutes of Standards and Technology*) pada tanggal 26 November 2001. Persyaratan yang diajukan oleh NIST terhadap algoritma baru tersebut adalah [14]:

1. Algoritma termasuk ke dalam kelompok algoritma kunci simetris berbasis *cipher* blok.
2. Seluruh rancangan algoritma tidak dirahasiakan.
3. Panjang kunci yang fleksibel: 128, 192, dan 256 *bit*.
4. Ukuran blok yang dienkripsi adalah 128 *bit*.
5. Algoritma dapat diimplementasikan dalam perangkat keras (*hardware*) maupun perangkat lunak (*software*).

Algoritma Rijndael menggunakan substitusi, permutasi, dan sejumlah putaran yang dikenakan pada setiap blok yang akan dienkripsi atau dekripsi. Di dalam setiap putarannya menggunakan kunci yang berbeda yang disebut sebagai *round key*.

2.8 Penelitian Terkait

Dilakukan penelitian dengan menggunakan perangkat lunak bernama *AESEncryptor* yang dibangun untuk implementasi algoritma kriptografi AES dengan mode operasi ECB, CBC, CFB, dan OFB untuk panjang kunci 128-*bit*, 192-*bit*, dan 256-*bit*. Algoritma AES merupakan algoritma Rijndael yang khusus berjalan pada ukuran blok 128-*bit*. *AESEncryptor* ini kemudian digunakan untuk membandingkan tingkat keamanan data algoritma AES pada berbagai mode operasi. Pengujian dilakukan dengan beberapa proses manipulasi terhadap arsip hasil enkripsi seperti pengubahan satu *bit* atau lebih blok *cipherteks*, penambahan blok *cipherteks* semu, dan penghilangan satu atau lebih blok *cipherteks*, kemudian dilakukan dekripsi untuk dibandingkan hasilnya dengan *plainteks* asal. Hasil pengujian menunjukkan bahwa AES merupakan salah satu solusi yang baik untuk mengatasi masalah keamanan dan kerahasiaan data [4].

Analisa terhadap struktur dan desain dari Rijndael yang menjadi kandidat AES pengganti DES, berdasarkan tiga kriteria berikut :

a) ketahanan terhadap serangan;
b) kecepatan dan kepadatan *code* pada berbagai *platform*; dan
c) kesederhanaan desain;

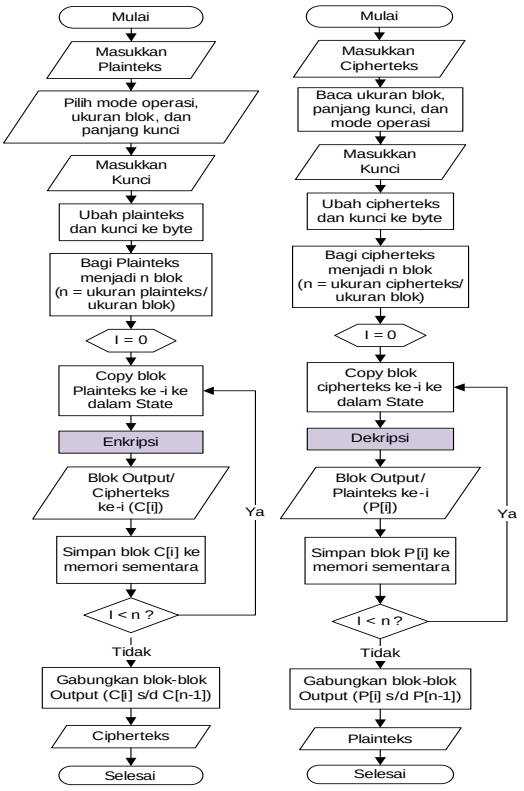
seperti kesamaan dan ketidaksamaannya dengan *cipher* simetris yang lainnya. Di sisi yang lain, penelitian ini juga meneliti kelebihan mendasar dari Rijndael dibandingkan DES dan *Triple* DES, begitu pula keterbatasannya. Sebagai contoh kenyataan bahwa *cipher* yang baru dan *invers*-nya menggunakan komponen yang berbeda, yang hampir menghilangkan kemungkinan kunci lemah dan kunci semi-lemah, yang ditemukan pada DES, dan juga ekspansi kunci nonlinier yang hampir menghilangkan kemungkinan kunci yang ekuivalen. Keduanya merupakan keuntungan mendasar dari Rijndael [11].

Algoritma Rijndael memiliki sejumlah keunggulan yang menyebabkannya dianggap layak menggantikan DES dan turunannya, seperti *Triple*-DES (3DES), salah satunya adalah tahan terhadap serangan XSL (*eXtended Sparse Linearisation*), sebuah teknik kriptanalisis terhadap *cipher* blok dengan sistem persamaan *overdefined* oleh Nicolas T. Courtois dan Josef Pieprzyk. Meskipun serangan yang berhasil dilakukan terhadap DES merupakan serangan yang mahal dan secara praktis tidak akan dilakukan dalam kehidupan sehari-hari sehingga banyak pihak yang tetap berani memberikan klaimnya mengenai keamanan yang ditawarkan, namun algoritma Rijndael memenangkan pertempuran keamanan secara mutlak bahkan jika dibandingkan dengan 3DES [12].

3. METODE PENELITIAN

3.1 Perancangan dan Implementasi Sistem

Secara singkatnya, langkah-langkah enkripsi data ditunjukkan oleh Gbr. 5(a) dan langkah-langkah dekripsi data ditunjukkan oleh Gbr. 5(b) berikut ini.



Gbr. 5 Diagram alir (a) enkripsi dan (b) dekripsi data

Penjelasan lebih detail proses enkripsi ditunjukkan oleh Gbr. 6(a) dan proses dekripsi ditunjukkan oleh Gbr. 6(b) berikut.

(a) (b)

Gambar 6. Proses (a) enkripsi dan (b) dekripsi

Beikut ini keterangan dari gambar 6 :

Enkripsi

a. Transformasi SubBytes
   Memetakan setiap *byte* dari *array state* dengan menggunakan tabel substitusi *S-Box*. Koordinat x merepresentasikan digit pertama dari bilangan heksadesimal, dan koordinat y merepresentasikan digit kedua. Misalnya bilangan heksadesimal 08, maka 0 berada di koordinat x dan 8 berada di koordinat y.
b. Transformasi ShiftRows
   Melakukan pergeseran secara *wrapping* (siklik) pada tiga baris terakhir dari *array state*. Jumlah pergeseran bergantung pada nilai baris r. Baris pertama, r = 0, tidak digeser. Baris-baris selanjutnya mengikuti persamaan berikut:
   $S'_{r,c} = S_{r,(c + shift(r, Nb)) \bmod Nb}$ untuk $0<r<4$; $0 \le c < Nb$ dimana nilai pergeseran *shift(r, Nb)* tergantung pada jumlah baris (*r*).
c. Transformasi MixColumns
   Mengalikan setiap kolom dari *array state* dengan polinom **$a(x) \bmod (x^4+1)$**. Secara lebih jelas, transformasi *MixColums* dapat dilihat pada perkalian matriks persamaan (i) dan (ii) berikut ini:

   $s'(x) = a(x) \quad s(x)$ (i)

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \quad \text{untuk } 0 \le c < Nb \quad \text{(ii)}$$

d. Transformasi AddRoundKey
   Melakukan operasi XOR terhadap *round key* dengan *array state*, hasilnya disimpan dalam *array state*.

Dekripsi

a. Transformasi InvSubBytes
   Memetakan setiap *byte* dari *array state* dengan menggunakan tabel substitusi kebalikan (*inverse S-Box*). Caranya sama dengan transformasi *SubBytes*.
b. Transformasi InvShiftRows
   Merupakan kebalikan dari transformasi *ShiftRow*. Transformasi ini juga melakukan pergeseran secara *wrapping* (siklik) pada tiga baris terakhir dari *array state*. Caranya sama dengan transformasi *ShiftRows*, hanya saja arah pergeserannya berlawanan (ke kanan).
c. Transformasi MixColumns
   Mengalikan setiap kolom dari *array state* dengan polinom **$a^{-1}(x) \bmod (x^4+1)$.** Secara lebih jelas, transformasi *InvMixColums* dapat dilihat pada perkalian matriks pada persamaan (iii) (iv) berikut ini:

   $s'(x) = a^{-1}(x) \quad s(x)$ (iii)

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{untuk } 0 \le c < Nb \quad \text{(iv)}$$

d. Transformasi AddRoundKey
   Transformasi ini sama dengan transformasi *AddRoundKey* pada enkripsi, yaitu dengan melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya disimpan dalam *array state*.

3.2 Pengujian dan Analisis

Pengujian dilakukan dengan menjalankan aplikasi untuk mendapatkan data waktu eksekusi proses enkripsi dan dekripsi terhadap *file-file* yang telah dipilih sebagai sampel. Selanjutnya dilakukan analisis dari hasil pengujian yang telah dilakukan. Secara teknis, tahapan ini dilakukan dengan cara sebagai berikut:

- Pengujian dilakukan pada empat berkas berukuran 2,5 MB, 5 MB, 10 MB, dan 20 MB.
- Masing-masing berkas dienkripsi dengan berbagai kombinasi panjang kunci, ukuran blok, dan mode operasi. Variasi panjang kunci dan ukuran blok adalah

128, 192, dan 256-*bit*, serta variasi mode operasi adalah ECB, CBC, dan CFB, (total ada 27 kombinasi).
- Dari keseluruhan proses enkripsi yang dilakukan terhadap masing-masing berkas dicatat waktunya.
- Setiap kombinasi dilakukan perulangan sebanyak lima kali (diambil waktu eksekusi rata-rata).
- Data yang diperoleh kemudian dianalisa bagaimana pengaruh variasi panjang kunci, ukuran blok, dan mode operasi terhadap waktu eksekusi.
- Pengambilan data waktu eksekusi dan analisa juga dilakukan terhadap proses dekripsi.

4. PEMBAHASAN

Pengujian waktu eksekusi dilakukan untuk mengetahui pengaruh panjang kunci, mode operasi, dan ukuran blok terhadap waktu eksekusi. Panjang kunci yang semakin besar akan meningkatkan keamanan data karena data yang terenkripsi akan semakin sulit untuk dibuka secara paksa (*brute force attack*). Ukuran panjang kunci dan ukuran blok mempengaruhi jumlah putaran (*ronde*) yang dilakukan oleh algoritma Rijndael, ditunjukkan di tabel 1.

Tabel 1. Jumlah Ronde terhadap Panjang Kunci dan Ukuran Blok

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

Nk adalah panjang kolom matriks kunci, *Nb* adalah panjang kolom matriks blok berkas, *Nr* adalah jumlah *ronde* yang merupakan fungsi dari *Nk* dan *Nb*. Setiap *state* kunci terdiri atas *Nk* x 4 x 1 *byte*, dan setiap *state* blok terdiri atas *Nb* x 4 x 1 *byte*. Jika *Nk* = 4, maka apabila diubah ke dalam *bit* menjadi 4 x 4 x 8 *bit* = 128 *bit*. *Nk* atau *Nb* bervariasi antara 4, 6, dan 8 yang apabila diubah ke dalam *bit* berturut-turut menjadi 128, 192, dan 256 *bit*.

Pengujian kecepatan enkripsi dan dekripsi dilakukan pada komputer dengan spesifikasi *processor* Intel(R) Pentium(R) *dual-core* T3200 (2 GHz, 667 MHz PSB, 1 MB L2 *cache*) dan 1 GHz DDR2. Pengujian dilakukan pada empat berkas pdf dengan ukuran 2,5 MB (2.579.713 *bytes*), 5 MB (5.136.388 *bytes*), 10 MB (11.053.969 *bytes*), dan 20 MB (21.138.955 *bytes*).

4.1 Hasil Pengujian Waktu Eksekusi Enkripsi pada Kunci 128-bit

Pengujian enkripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan Tabel 2.

Tabel 2. Waktu Eksekusi Rata-rata Enkripsi Berkas 2,5 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.478125	0.490625	0.522000
CBC	0.503125	0.546875	0.568750
CFB	0.521875	0.615625	0.753125

Pengujian enkripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 3.

TABEL 3. Waktu Eksekusi Rata-rata Enkripsi Berkas 5 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.815625	0.837500	0.871875
CBC	0.853125	0.862500	0.881250
CFB	0.887500	1.075000	1.268750

Pengujian enkripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 4.

Tabel 4. Waktu Eksekusi Rata-rata Enkripsi Berkas 10 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.634375	1.684375	1.762500
CBC	1.775000	1.778125	1.803125
CFB	1.784375	2.134375	2.696875

Pengujian enkripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 5.

Tabel 5. Waktu Eksekusi Rata-rata Enkripsi Berkas 20 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	3.159375	3.196875	3.318750
CBC	3.234375	3.300000	3.378125
CFB	3.309375	4.150000	5.118730

Dari Tabel 2 s/d Tabel 5 di atas terlihat pada mode ECB, enkripsi membutuhkan waktu paling sedikit, selanjutnya CBC, dan CFB membutuhkan waktu paling banyak. Dengan variasi ukuran blok, semakin besar ukuran blok semakin lama waktu yang dibutuhkan. Hal ini karena pada kunci 128-*bit*, semakin besar ukuran blok semakin banyak jumlah putaran enkripsi. Blok 128-*bit* memiliki 10 putaran enkripsi, blok 192-*bit* memiliki 12 putaran enkripsi, dan blok 256-*bit* memiliki 14 putaran enkripsi.

4.2 Hasil Pengujian Waktu Eksekusi Dekripsi pada Kunci 128-bit

Pengujian dekripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 6.

Tabel 6. Waktu Eksekusi Rata-rata Dekripsi Berkas 2,5 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.665625	0.690625	0.803125
CBC	0.671875	0.700000	0.840625
CFB	0.737500	0.768750	1.000000

Pengujian dekripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 7.

Tabel 7. Waktu Eksekusi Rata-rata Dekripsi Berkas 5 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.206250	1.268750	1.315625
CBC	1.225000	1.275000	1.387500
CFB	1.296875	1.446875	1.703125

Pengujian dekripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 8.

Tabel 8. Waktu Eksekusi Rata-rata Dekripsi Berkas 10 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	2.503125	2.631250	2.712500
CBC	2.546875	2.665625	2.743675
CFB	2.700000	3.034375	4.021875

Pengujian dekripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 128-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 9.

Tabel 9. Waktu Eksekusi Rata-rata Dekripsi Berkas 20 MB Kunci 128-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	4.746875	4.793750	4.931250
CBC	4.753125	4.800000	5.000000
CFB	4.896875	5.696875	6.703125

Dari Tabel 6 s/d Tabel 9 dapat dilihat hasil pengujian waktu eksekusi dekripsi dengan berbagai kombinasi mode operasi dan ukuran blok pada kunci 128-*bit* menunjukkan hasil yang bersesuaian dengan hasil pada proses enkripsi.

4.3 Hasil Pengujian Waktu Eksekusi Enkripsi pada Kunci 192-bit

Pengujian enkripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 10.

Tabel 10. Waktu Eksekusi Rata-rata Enkripsi Berkas 2,5 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.506250	0.503125	0.528125
CBC	0.521875	0.518750	0.553125
CFB	0.553125	0.593750	0.725000

Pengujian enkripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 11.

Tabel 11. Waktu Eksekusi Rata-rata Enkripsi Berkas 5 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.853125	0.831250	0.884375
CBC	0.878125	0.871875	0.909375
CFB	0.921875	1.053125	1.321875

Pengujian enkripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 12.

Tabel 12. Waktu Eksekusi Rata-rata Enkripsi Berkas 10 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.750000	1.743750	1.771875
CBC	1.803125	1.787500	1.818750
CFB	1.881250	2.106250	2.646875

Pengujian enkripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 13.

Tabel 13. Waktu Eksekusi Rata-rata Enkripsi Berkas 20 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	3.228125	3.225000	3.309375
CBC	3.346875	3.343750	3.403125
CFB	3.446875	4.128125	4.931250

Dari Tabel 10 s/d Tabel 13 terlihat proses enkripsi paling cepat adalah pada mode ECB, dilanjutkan CBC, dan CFB membutuhkan waktu paling lama. Pada kunci 192-*bit*, blok 128-*bit* dan 192-*bit* sama-sama memiliki 12 putaran enkripsi, sedang blok 256-*bit* memiliki 14 putaran enkripsi. Pada ECB dan CBC, meskipun jumlah putarannya sama, namun blok 128-*bit* memiliki waktu eksekusi yang lebih lama dibanding blok 192-*bit*. Hal ini karena blok 128-*bit* memiliki jumlah pecahan *state* yang lebih banyak dibandingkan blok 192-*bit*, sehingga membutuhkan waktu yang lebih banyak untuk menggabungkannya. Blok 256-*bit* memiliki waktu eksekusi paling lama karena jumlah putaran enkripsinya paling banyak. Pada mode CFB terjadi hal yang berbeda, yaitu semakin besar ukuran blok semakin lama waktu eksekusi. Hal ini karena pada mode CFB jumlah putaran relatif tidak mempengaruhi waktu eksekusi. Waktu eksekusi relatif dipengaruhi oleh panjang antrian yang berukuran sama dengan ukuran blok masukan, sehingga semakin besar ukuran blok, semakin panjang pula antrian. Hal ini mengakibatkan proses enkripsi membutuhkan waktu yang semakin lama.

4.4 Hasil Pengujian Waktu Eksekusi Dekripsi pada Kunci 192-bit

Pengujian dekripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 14.

Tabel 14. zdvnkWaktu Eksekusi Rata-rata Dekripsi Berkas 2,5 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.684375	0.681250	0.721875
CBC	0.700000	0.690625	0.731250
CFB	0.753125	0.778125	0.937500

Pengujian dekripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 15.

Tabel 15. Waktu Eksekusi Rata-rata Dekripsi Berkas 5 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.281250	1.271875	1.318750
CBC	1.296875	1.278125	1.328125
CFB	1.359375	1.443750	1.668750

Pengujian dekripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 16.

Tabel 16. Waktu Eksekusi Rata-rata Dekripsi Berkas 10 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	2.653125	2.609375	2.706250
CBC	2.684375	2.662500	2.850000
CFB	2.771875	3.053125	3.553125

Pengujian dekripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 192-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 17.

Tabel 17. Waktu Eksekusi Rata-rata Dekripsi Berkas 20 MB Kunci 192-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	4.875000	4.865625	4.978125
CBC	4.896875	4.868750	5.465625
CFB	5.056250	5.665625	6.593750

Dari Tabel 14 s/d Tabel 17 terlihat hasil pengujian waktu eksekusi dekripsi yang diperoleh dari berbagai kombinasi mode operasi dan ukuran blok untuk kunci 192-*bit* menunjukkan hasil yang bersesuaian dengan hasil pada proses enkripsi.

4.5 Hasil Pengujian Waktu Eksekusi Enkripsi pada Kunci 256-bit

Pengujian enkripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 18.

Tabel 18. Waktu Eksekusi Rata-rata Enkripsi Berkas 2,5 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.546875	0.534375	0.531250
CBC	0.584375	0.575000	0.571875
CFB	0.621875	0.643750	0.734375

Pengujian enkripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 19.

Tabel 19. Waktu Eksekusi Rata-rata Enkripsi Berkas 5 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.909375	0.881250	0.878125
CBC	0.921875	0.915625	0.884375
CFB	0.978125	1.118750	1.293750

Pengujian enkripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 20.

Tabel 20. Waktu Eksekusi Rata-rata Enkripsi Berkas 10 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.840625	1.803125	1.784375
CBC	1.946875	1.887500	1.818750
CFB	2.168750	2.312500	2.656250

Pengujian enkripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 21.

Tabel 21. Waktu Eksekusi Rata-rata Enkripsi Berkas 20 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	3.434375	3.243750	3.231250
CBC	3.584375	3.428125	3.365625
CFB	3.843750	4.290625	5.056250

Dari Tabel 18 s/d Tabel 21 di atas terlihat pada mode ECB, enkripsi membutuhkan waktu paling sedikit, selanjutnya CBC, dan CFB membutuhkan waktu paling banyak. Pada ECB dan CBC, semakin besar ukuran blok waktu yang dibutuhkan semakin sedikit, sedangkan pada CFB terjadi sebaliknya, yaitu semakin besar ukuran blok waktu yang dibutuhkan semakin banyak. Hal ini karena pada mode ECB dan CBC kunci 256-*bit*, variasi ukuran blok tidak mempengaruhi jumlah putaran. Seperti terlihat pada Tabel 2, ketiga variasi ukuran blok sama-sama memiliki 14 putaran. Kecenderungan waktu eksekusi yang semakin kecil untuk ukuran blok yang semakin besar, hal ini karena semakin besar ukuran blok, jumlah pecahan *state* semakin sedikit, sehingga waktu yang diperlukan untuk menggabungkannya juga semakin sedikit. Sedangkan pada mode CFB, lamanya waktu eksekusi relatif dipengaruhi oleh adanya antrian yang berukuran sama dengan ukuran blok masukan, dimana semakin besar ukuran blok, semakin panjang pula antriannya, sehingga waktu eksekusinya juga semakin lama.

4.6 Hasil Pengujian Waktu Eksekusi Dekripsi pada Kunci 256-bit

Pengujian dekripsi berkas 2,5 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-bit menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 22.

Tabel 22. Waktu Eksekusi Rata-rata Dekripsi Berkas 2,5 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	0.750000	0.721875	0.712500
CBC	0.753125	0.731250	0.728125
CFB	0.800000	0.878125	0.931250

Pengujian dekripsi berkas 5 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 23.

Tabel 23. Waktu Eksekusi Rata-rata Dekripsi Berkas 5 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	1.350000	1.300000	1.303125
CBC	1.375000	1.359375	1.350000
CFB	1.468750	1.534375	1.725000

Pengujian dekripsi berkas 10 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 24.

Tabel 24. Waktu Eksekusi Rata-rata Dekripsi Berkas 10 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	2.906250	2.796875	2.709375
CBC	2.946875	2.828125	2.753125
CFB	3.068750	3.268750	3.584375

Pengujian dekripsi berkas 20 MB dengan variasi mode operasi dan ukuran blok pada kunci 256-*bit* menghasilkan rata-rata waktu eksekusi yang ditunjukkan oleh Tabel 25.

Tabel 25. Waktu Eksekusi Rata-rata Dekripsi Berkas 20 MB Kunci 256-bit

Mode | Nb	128-bit (s)	192-bit (s)	256-bit (s)
ECB	5.293750	5.137500	4.990625
CBC	5.387500	5.234375	5.181250
CFB	5.706250	6.253125	6.756250

Dari Tabel 22 s/d Tabel 25 terlihat hasil pengujian waktu eksekusi dekripsi yang diperoleh dari berbagai kombinasi mode operasi dan ukuran blok untuk panjang kunci 256-*bit* menunjukkan hasil yang bersesuaian dengan hasil pada proses enkripsi.

Dari keseluruhan data hasil pengujian waktu eksekusi di atas dapat disimpulkan bahwa:

- Pada mode ECB dan CBC berlaku sebagai berikut:
  - Pada kunci 128-bit: semakin besar ukuran blok (Nb), semakin banyak jumlah putaran enkripsi/dekripsi, sehingga waktu eksekusi semakin lama.
  - Pada kunci 192-bit: semakin besar ukuran blok (pada blok 128-bit dan blok 192-bit), dengan jumlah putaran sama (12 putaran), semakin kecil waktu eksekusinya, karena jumlah pecahan *state* yang harus digabungkan semakin sedikit. Namun pada blok 256-bit, karena jumlah putarannya paling banyak (14 putaran) maka waktu eksekusinya pun paling lama.
  - Pada kunci 256-bit: ukuran blok tidak mempengaruhi jumlah putaran (sama-sama memiliki 14 putaran). Semakin besar ukuran blok, semakin sedikit jumlah pecahan *state* yang harus digabungkan, sehingga waktu eksekusi juga semakin cepat.
  - Pada mode CFB berlaku: waktu eksekusi relatif tidak dipengaruhi oleh jumlah putaran enkripsi/dekripsi melainkan relatif dipengaruhi oleh ukuran blok. Semakin besar ukuran blok semakin lama waktu eksekusi. Hal ini dikarenakan adanya sebuah antrian yang panjangnya sama dengan ukuran blok.

5. PENUTUP

Dari hasil pengujian dan analisa algoritma Rijndael dapat diambil kesimpulan sebagai berikut:

- Urutan lama waktu yang digunakan untuk proses enkripsi dan dekripsi algoritma Rijndael dengan mode operasi ECB, CBC, dan CFB secara berturut-turut mulai dari yang tercepat adalah: ECB – CBC – CFB.
- Pada kunci 128-bit, semakin besar ukuran blok semakin lama waktu eksekusi. Waktu eksekusi paling kecil adalah pada mode ECB dengan blok 128-bit, sedangkan waktu eksekusi paling lama adalah pada mode CFB dengan blok 256-bit.
- Kecepatan eksekusi pada mode ECB dan CBC relatif dipengaruhi oleh jumlah putaran, dimana jumlah putaran tergantung pada panjang kunci dan ukuran blok. Pada mode ECB dan CBC, semakin banyak jumlah putaran semakin lama waktu eksekusi. Apabila jumlah putaran sama, semakin besar ukuran blok semakin cepat waktu eksekusi.
- Kecepatan eksekusi pada mode CFB relatif dipengaruhi oleh ukuran blok, semakin besar ukuran blok semakin lama waktu eksekusi.

6. DAFTAR PUSTAKA

[1] Delfianto, R. 2010. *Studi dan Perbandingan Algoritma Rijndael dengan Algoritma Serpent*. Bandung: Program Studi Teknik Informatika ITB.

[2] Viqarunnisa, P. 2006. *Studi dan Perbandingan Algoritma Rijndael dan Twofish*. Bandung: Program Studi Teknik Informatika ITB.

[3] Ilmy, M.B. 2006. *Perbandingan Algoritma Mars dan Rijndael dalam Beberapa Mode Operasi Cipher Blok*. Bandung: Program Studi Teknik Informatika ITB.

[4] Lung, C. 2004. *Studi dan Implementasi Advanced Encryption Standard dengan Empat Mode Operasi Block Cipher*. Makalah Seminar Tugas Akhir. Bandung: Departemen Teknik Informatika ITB.

[5] Daemen, J., and Rijmen, V. 1999. *AES Proposal: Rijndael*. *AES Algorithm Submission*, September 3, 1999.

[6] Yusuf, V.R. 2007. *Aplikasi Enkripsi dan Dekripsi Menggunakan Rijndael*. Makalah Seminar Tugas Akhir. Semarang: Jurusan Teknik Elektro FT UNDIP.

[7] Schneier, B. 1996. *Applied Cryptography - Protocols, Algorithms and Source Code in C*, John Wiley & Sons Inc., 2nd Edition.

[8] Menezes, A.J., van Oorschot, P.C., and Vanstone, S.A. 1996. *Handbook of Applied Cryptography*. Penerbit CRC Pres.

[9] Walton, J. 2008. *Applied Crypto++: Block Ciphers*. (Online),(http://www.codeproject.com/KB/security/BlockCiphers.aspx, diakses 21 September 2011).

[10] Munir, R. 2006. *Kriptografi*. Bandung: Penerbit Informatika.

[11] Avila, C.S., and Reillo, R.S. 2001. The Rijndael Block Cipher (AES Proposal): A Comparison with DES. *Proceeding IEEE 35th Annual 2001 International Carnahan Conference on Security Technology*, October 16-19, 2001, London, UK.

[12] Handaka, M.S. 2011. *Analisis AES Rijndael terhadap DES*. Bandung: Program Studi Teknik Informatika ITB .