YAYASAN AKRAB PEKANBARU
Jurnal AKRAB JUARA
Volume 4 Nomor 3 Edisi Agustus 2019 (1-10)

21

IMPLEMENTASI FULL DISK ENCRYPTION DENGAN ALGORITMA AES-XTS MENGGUNAKAN VERACRYPT

Dewi Laksmiati
Universitas Bina Sarana Informatika
(Naskah diterima: 1 Juni 2019, disetujui: 28 Juli 2019)

Abstract

Data security is currently a major concern in the world of data communication systems. Especially for important and sensitive data that must be kept confidential from unauthorized parties. There are several ways that can be applied to secure the data communicated. This paper will explain the effective security design for data communication using the AES algorithm for encryption and decryption. The National Institute of Standards and Technology (NIST) initiated a process to develop one in the Advanced Encryption Standard (AES), which also established an Advanced Encryption Algorithm to replace the 1998 Data Encryption Standard (DES) algorithm. Advanced Encryption Standard (AES) can be implemented through programs on software or embedded in hardware. In continuous development, NIST develop XTS as an advance standard that applies to full disk encryption. In this paper we will discuss the application of AES-XTS using the Veracrypt encryption application

Keywords: *AES, XTS, Block Cipher, Cryptography, DES, NIST, Veracyrpt.*

Abstrak

Keamanan data saat ini merupakan perhatian utama dalam dunia sistem komunikasi data. Terutama data-data penting dan sensitif yang harus dirahasiakan dari pihak yang tidak berkepentingan. Ada beberapa cara yang dapat diterapkan untuk mengamankan data yang dikomunikasikan. Penulisan ini akan menjelaskan desain keamanan yang efektif untuk komunikasi dat menggunakan algoritma AES untuk enkripsi dan dekripsi. National Institute of Standards and Technology (NIST) telah menginisiasi proses untuk mengembangkan Advanced Encryption Standard (AES), yang juga menetapkan suatu Algoritma Enkripsi tingkat Lanjut untuk menggantikan algoritma Data Encryption Standard(DES) yang Kedaluwarsa pada tahun 1998. Advanced Encryption Standard (AES) dapat diterapkan melalui program pada perangkat lunak atau ditanamkan pada perangkat keras. Dalam perkembangannya NIST mengembagkan XTS sebagai merupakan standar lanjutan dalam enkripsi disk menyeluruh. Dalam penulisan ini akan dibahas penerapan enkripsi disk secara penuh dengan algoritma AES-XTS menggunakan aplikasi enkripsi Veracrypt.

Kata Kunci: AES, XTS, Block Cipher, Cryptography, DES, NIST, Veracyrpt.

I. PENDAHULUAN

Perkembangan teknologi komunikasi dan jaringan saat ini, memudahkan manusia sehingga dapat berkomunikasi dan saling bertukar data dan informasi dengan cepat dan tanpa dihalangi oleh jarak.

Dalam beberapa tahun terakhir keamanan dan integritas data perhatian utama. Dimana hampir semua data ditransfer melalui jaringan komputer dan juga data yang disimpan didalam media penyimpanan rentan terhadap berbagai macam serangan. Untuk membuat data aman dari berbagai serangan dan untuk integritas data kita harus mengenkripsi data sebelum dikirim atau disimpan. Ilmu yang mempelajari tentang cara-cara pengamanan data dengan penyandian dikenal dengan nama Kriptografi.

Kriptografi adalah metode penyimpanan dan transmisi data dimana hanya orang yang berhak dapat membaca dan memroses data. Kriptografi merupakan cabang ilmu yang digunakan untuk melindungi informasi asli dalam format plaintext dengan cara menyandikannya ke dalam format yang telah disandikan (ciphertext). Proses penyandian ini disebut dengan enkripsi (encryption). Proses sebaliknya, penerjemahan cipherteks menjadi plainteks disebut dengan proses dekripsi (decryption). Proses enkripsi dan dekripsi tersebut memakai satu atau lebih kunci kriptografi.

Proses enkripsi dan dekripsi dalam penelitian ini menggunakan AES standar VeraCrypt dengan 14 *rounds* and a 256-bit key (i.e., AES-256, published in 2001) yang beroperasi pada XTS mode (AES-XTS. Algoritma ini merupakan algoritma yang digunakan untuk mengamankan data yang disimpan pada enkripsi disk menyeluruh*(full disk encryption)*. Menggunakan aplikasi Veracrypt, sebuah software enkripsi dengan metode on the fly, dimana proses enkripsi dan dekripsi akan dilakukan secara langsung pada saat pembacaan dan pemrosesan data. Data yang dienkripsi akan ditampung pada file container yang hanya dapat dibaca menggunakan aplikasi Veracrypt dengan key yang tepat.

II. KAJIAN TEORI

2.1 AES (*Advanced Encryption Standard*)

AES pada awalnya dikembangkan untuk menggantikan algoritma enkripsi DES. Dimulai pada Januari 1997, NIST mulai mengembangkan AES sebagai algoritma enkripsi kunci simetris, dan mengumumkan keseluruh dunia algoritma sebagai pengganti DES.

Awalnya ada 15 algoritma yang menjadi calon pengganti DES, yang kemudian direduksi menjadi 4 algoritma, RC6, Rijndael,

Serpent dantwo-fish, yang semuanya adalah cipher blok iterated. Keempat algoritma tersebut dirancang untuk memenuhi syarat sebagai AES.

Penentuan untuk memilih yang terbaik dari keempat algoritma tersebut berdasarkan pada tiga karakteristik, yaitu:

1. Keamanan: Mencakup ketahanan terhadap serangan yang diketahui, kokoh secara matematis, keacakan output dan keamanan dibandingkan dengan algoritma lainnya.
2. Biaya: Mencakup kecepatan enkripsi, memori yang dibutuhkan,dan tidak ada perjanjian lisensi yaitu algoritma harus tersedia seluruh dunia secara gratis tanpa royalty.
3. Algoritma dan karakteristik implementasi: Algoritma harus sesuai di berbagai sistem perangkat keras dan perangkat lunak. Algoritma juga harus relatif sederhana.

Setelah tinjauan mendalam Algoritma Rijndael dipilih untuk menjadi algoritma AES. Di bawah ini perbedaan algoritma AES dengan pendahulunya algoritma DES.

Tabel 1
Perbedaan antara AES dan DES

Faktor	DES	AES
Penjang Key	56 bits	128, 192, 256 bits
Ukuran Blok	64 bits	128, 192, 256 bits
Cipher Text	Symmetric block cipher	Symmetric block cipher
Dikembangkan	1977	2000
Keamanan	Terbukti tidak memadai	Dianggap aman
Cryptanalysis resistance	Rentan terhadap kriptoanalisis diferensial dan linier	Kuat terhadap kriptoanalisis diferensial dan linier
Possible Keys	2^{56}	2^{128}, 2^{192}, 2^{256}

Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [2]. Perbedaan dari ketiga urutan tersebut adalah panjang kunci yang mempengaruhi jumlah round (perputaran) yang dapat digambarkan dalam bentuk table.

Tabel 2
Tabel urutan data algoritma AES

	Panjang Kunci	Panjang Blok	Jumlah Putaran
AES-128	4	4	10
AES-129	6	4	12
AES-256	8	4	14

Tabel 2 di atas menjelaskan tentang variasi algoritma AES dengan panjang kunci, panjang blok dan jumlah putaran yang berbeda-beda.

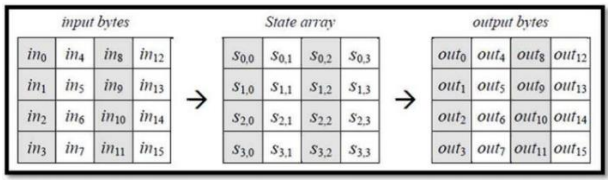
Terdapat 4 transformasi putaran/ *rounds* pada proses enkripsi dan dekripsi:

1. *SubBytes* Fungsinya menukar isi dari *byte* dengan memakai tabel substitusi
2. *ShiftRows* Proses pergeseran blok per baris pada *state array*.
3. *MixColumn* Proses mengalikan blok data (pengacakan) di masing-masing *state array* dengan rumus sebagai berikut:
   $A(x) = \{03\}x^2 + \{01\}x^2 + \{01\}x + \{02\}$
4. *AddRoundKey* Mengombinasikan state array dan round key dengan hubungan XOR.

Pada proses dekripsi algoritma AES prosesnya sebagai berikut:

1. *InvShiftRows*, Melakukan pergeseran bit ke kanan pada setiap blok baris.
2. *InvSubBytes* Setiap elemen pada *state* dipetakan dengan tabel *Inverse S-Box*.
3. *InvMixColumn* Setiap kolom dalam state dikalikan dengan matriks AES.
4. *AddRoundKey* Mengombinasikan *state array* dan *round key* dengan hubungan XOR.

Penggambaran proses transformasi putaran dapat dilihat dari Gambar 1.

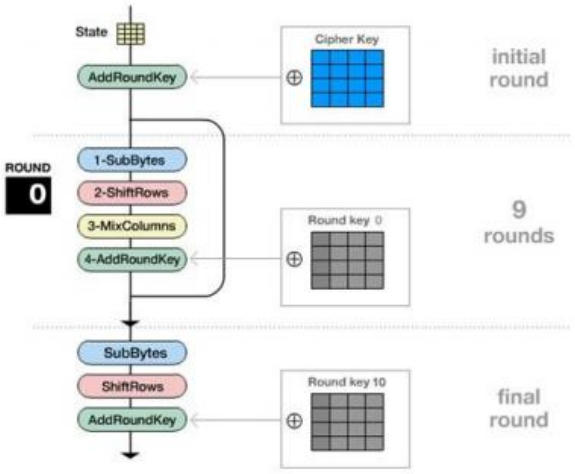


Gambar 1.
Proses *input bytes, state array, output bytes*

Dari Gambar 1 di atas menunjukkan bahwa algoritma AES memiliki dasar, dimana algoritma AES ini merupakan *array of bytes*, memiliki dua dimensi yang disebut dengan state. Rumus ukuran state yaitu $NROWS\ x\ NCOLS$, melalui state ini akan dilakukan enkripsi dan dekripsi yang kemudian hasilnya dimasukkan ke dalam *array of state*. Proses enkripsi dimulai dengan memasukkan data ke dalam *input bytes* yang kemudian disalin ke dalam *array state*, melalui proses ini kemudian dilakukan enkripsi serta dekripsi, hasil keluaran yang didapat akan ditampung dalam output bytes.

Di awal proses enkripsi, input yang tersalin dalam state akan mengalami transformasi *AddRoundKey*. Kemudian *state* akan mengalami transformasi *SubBytes, ShiftRows, Mix Columns,* dan *AddRoundKey* secara berulang sebanyak *round*/putaran (Nr). Proses dalam algoritma AES ini disebut sebagai *round function*. Pada *round* atau putaran yang terakhir, *state* tidak diberikan transformasi *MixColumns* [5]. Ilustrasi pemrosesan awal pada en-

kripsi menggunakan algoritma AES -128 dapat dilihat pada Gambar 2.

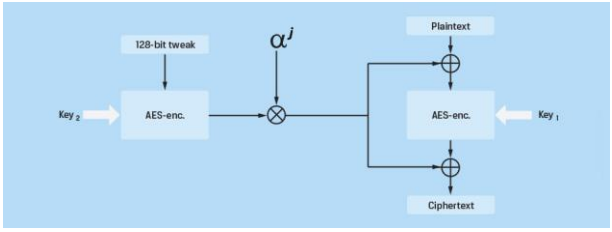


Gambar 2.
Contoh proses enkripsi dengan menggunakan algoritma AES-128

2.2 XTS(*ciphertext stealing)*

XTS yang dimaksud adalah Mode Blok Cipher AES-XTS. XTS awalnya ditetap-kan sebagai IEEE Std 1619-2007, NIST me-nambahkan XTS ke daftar mode blok cipher AES pada 2010. XTS dirancang sebagai alter-natif yang lebih kuat untuk mode cipher blok lain yang tersedia seperti CBC. Ini menghi-langkan potensi kerentanan yang terkait de-ngan beberapa serangan yang lebih canggih yang dapat digunakan untuk mengeksploitasi kelemahan dalam mode enkripsi yang lain.

Gambar 3 adalah diagram blok yang disederhanakan untuk mode XTS[8]

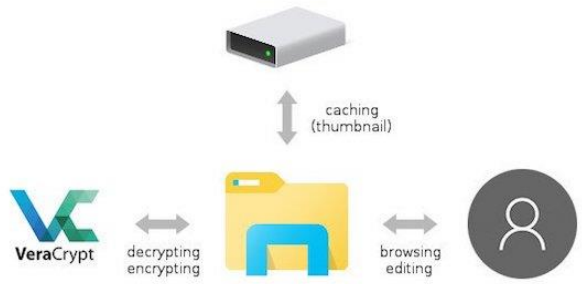


Gambar 3.
Pemrosesan Enkripsi disk AES-XTS

XTS menggunakan dua kunci AES. Satu kunci digunakan untuk melakukan enkripsi blok AES, kunci yang lain digunakan untuk mengenkripsi apa yang dikenal sebagai "Nilai Tweak." Tweak terenkripsi ini selanjutnya dimodifikasi dengan fungsi polinomial Galois / *Galois polynomial Function* (GF) dan XOR dengan teks biasa dan teks sandi dari setiap blok. Fungsi GF memberikan difusi lebih lan-jut dan memastikan bahwa blok data yang identik tidak akan menghasilkan teks sandi yang identik. Ini mencapai tujuan setiap blok menghasilkan teks sandi unik yang diberikan teks biasa identik tanpa menggunakan vektor inisialisasi dan rantai. Akibatnya, teks mende-kati dienkripsi ganda menggunakan dua kunci independen. Dekripsi data dilakukan dengan membalikkan proses ini. Karena setiap blok independen dan tidak ada rantai, jika data sandi yang disimpan rusak dan menjadi rusak, hanya data untuk blok tertentu yang tidak dapat dipulihkan. Dengan mode rantai, kesala-han ini dapat menyebar ke blok lain saat di-dekripsi.

2.3 Veracrypt

VeraCrypt adalah perangkat lunak untuk membuat dan memelihara volume(perangkat penyimpanan data) terenkripsi secara *on-the-fly* . Dimana VeraCrypt membuat sebuah volume penyimpanan atau dapat juga mengenkripsi keseluruhan volume disk. Kemudian data dienkripsi dan dideskripsi secara *on-the-fly*. Enkripsi secara *on-the-fly* berarti data dienkripsi secara otomatis tepat sebelum disimpan dan didekripsi tepat setelah diambil datanya, tanpa campur tangan pengguna. Seluruh sistem file dalam volume penyimpanan dienkripsi (misalnya., nama file, nama folder, konten setiap file, ruang kosong, meta data, dll).



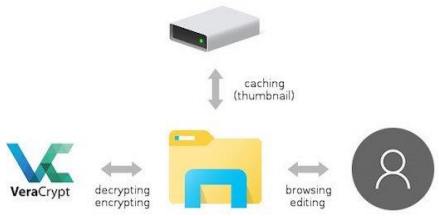
Gambar 3.
Pemrosesan Enkripsi disk AES-XTS

XTS menggunakan dua kunci AES. Satu kunci digunakan untuk melakukan enkripsi blok AES, kunci yang lain digunakan untuk mengenkripsi apa yang dikenal sebagai "Nilai Tweak." Tweak terenkripsi ini selanjutnya dimodifikasi dengan fungsi polinomial Galois / *Galois polynomial Function* (GF) dan XOR dengan teks biasa dan teks sandi dari setiap blok. Fungsi GF memberikan difusi lebih lanjut dan memastikan bahwa blok data yang identik tidak akan menghasilkan teks sandi yang identik. Ini mencapai tujuan setiap blok menghasilkan teks sandi unik yang diberikan teks biasa identik tanpa menggunakan vektor inisialisasi dan rantai. Akibatnya, teks mendekati dienkripsi ganda menggunakan dua kunci independen. Dekripsi data dilakukan dengan membalikkan proses ini. Karena setiap blok independen dan tidak ada rantai, jika data sandi yang disimpan rusak dan menjadi rusak, hanya data untuk blok tertentu yang tidak dapat dipulihkan. Dengan mode rantai, kesalahan ini dapat menyebar ke blok lain saat didekripsi.

2.4 Veracrypt

VeraCrypt adalah perangkat lunak untuk membuat dan memelihara volume(perangkat penyimpanan data) terenkripsi secara *on-the-fly* . Dimana VeraCrypt membuat sebuah volume penyimpanan atau dapat juga mengenkripsi keseluruhan volume disk. Kemudian data dienkripsi dan dideskripsi secara *on-the-fly*. Enkripsi secara *on-the-fly* berarti data dienkripsi secara otomatis tepat sebelum

disimpan dan didekripsi tepat setelah diambil datanya, tanpa campur tangan pengguna. Seluruh sistem file dalam volume penyimpanan dienkripsi (misalnya., nama file, nama folder, konten setiap file, ruang kosong, meta data, dll).



Gambar 4.
Proses enkripsi dan dekripsi menggunakan VeraCrypt

Volume data terenkripsi dapat dibaca (didekripsi) menggunakan VeraCrypt untuk dibuka, tahap awal adalah *mounting* volume data menggunakan kata sandi / *keyfile* atau kunci enkripsi. Setelah volume data terbuka, maka volume data akan tampil sebagaimana sebuah drive normal. File dapat disalin dari dan ke volume penyimpanan yang dibuat VeraCrypt sama seperti mereka disalin dari dan ke disk normal apa pun (misalnya, proses kopi file secara *drag-and-drop*). File secara otomatis didekripsi dengan cepat (dalam memori / RAM) saat sedang dibaca atau disalin dari volume VeraCrypt yang dienkripsi. Demi--kian pula, file yang sedang ditulis atau disalin ke volume VeraCrypt secara otomatis dienkripsi dengan cepat (tepat sebelum ditulis ke disk) dalam RAM. Harap diperhatikan bahwa ini tidak berarti bahwa seluruh file yang akan dienkripsi/didekripsi harus disimpan dalam RAM sebelum dapat dienkripsi / didekripsi. Tidak ada persyaratan memori tambahan (RAM) untuk VeraCrypt. VeraCrypt mampu mengakomodir berbagai macam enkripsi AES, Serpent, TwoFish dan beberapa algoritma Enkripsi.

III. METODE PENELITIAN

Melakukan pengumpulan data-data dengan cara mengamati serta mencatat secara sistematis tentang perangkat dan aplikasi yang digunakan dalam konfigurasi dalam prak-tek langsung.

Yaitu menggunakan literature baik dalam bentuk media online, artikel atau buku bacaan yang berkaitan dengan penyusunan artikel ini.

3.1 Metode Pengembangan Jaringan

1. Analisa Kebutuhan

Analisa akan dilakukan melalui beberapa tahapan, yaitu:

a. Observasi langsung
b. Memahami semua kondisi kebutuhan di lapangan terkait *Full Disk Encryption*

c. Analisia hasil observasi

2. Desain

Perancangan akan dilakukan melalui beberapa tahapan, yaitu:

a. Pemilihan algoritma enkripsi menyesuaikan kebutuhan
b. Pemilihan hardware penyimpanan yang ideal untuk pengujian

3. Testing

Melakukan uji coba langsung menggunakan *PC* yang sudah terinstall VeraCrypt.

4. Instalasi dan Enkripsi

Untuk menjalankan VeraCrypt diperlukan instalasi sebagai berikut:

a. Instalasi VeraCrypt pada Windows

VeraCrypt tersedia untuk Free BSD, Linux, Mac, dan Windows. Dalam pengujian ini, VeraCrypt yang digunakan versi windows dengan pertimbangan lebih user-friendly dan proses instalasi yang lebih mudah.

Spesifikasi hardware yang digunakan:

Processor : Intel Core i5-5300U
RAM : 4GB

b. Format dan Enkripsi Awal Flashdisk Menggunakan Veracrypt

Setelah instalasi VeraCrypt, selanjutnya dilakukan pembuatan partisi terenkripsi pada flashdisk yang terlah ditentukan dengan spesifikasi berikut :

Kapasitas : 8GB
USB Mode : 2.0
Format : exFAT

Proses format dan enkripsi menghabiskan waktu 18 menit.

IV. HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil pengujian dan analisa pembandingan pengukuran kecepatan operasi disk saat dengan dan tanpa enkripsi.

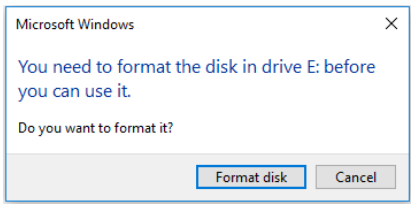
4.1 Pengujian Buka dan Deteksi Disk Terenkripsi Tanpa VeraCrypt

Pada sub bab ini akan dijelaskan mengenai hasil pengujian dan analisa disk yang telah terenkripsi menggunakan beberapa metode sederhana dan tools pendeteksi enkripsi.

Melalui beberapa pengujian system tidak bias dibaca dan dideteksi melalui metode normal.

a. **Pengujian Buka Melalui Windows Explorer dan Windows disk Management**

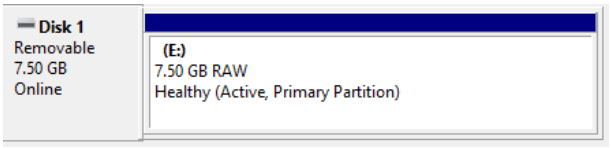
Pada Windows Explorer, flashdisk terenkripsi terdeteksi sebagai flashdisk yang belum terformat.



Gambar 5.

Flashdisk terenkripsi terbaca butuh diformat

Sedangkan pada tools Windows Disk Management, partisi flashdisk tidak terdeteksi dan terindikasi sebagai *RAW Partition*



Gambar 6.

Flashdisk terenkripsi terbaca sebagai *RAW Partition*

b. **Pengujian Deteksi Menggunakan Encrypted Disk Detector(EDD)**

Pengujian selanjutnya menggunakan aplikasi Encrypted Disk Detector, hasil yang didapatkan seperti di bawah ini:

```
Encrypted Disk Detector v2.2.1
Copyright (c) 2009-2019 Magnet Forensics Inc.
http://www.magnetforensics.com

<output omitted>

* Now checking logical volumes on system... *

Drive C: is located on PhysicalDrive0,
Partition #4.
Drive D: is located on PhysicalDrive0,
Partition #5.
```

	Kecepatan Rata-rata (MB/s)		
	Tanpa Enkripsi	Dengan Enkripsi	Perbedaan
Read	24,74	25,04	1,21 %
Write	8,961	9,36	4,45%

```
Drive E: is located on PhysicalDrive1,
Partition #1.
Drive F: appears to be a virtual disk
        - possibly a TrueCrypt or PGP
encrypted volume
```

Pengujian di atas dilakukan setelah partisi terenkripsi dimounting ke drive F:. Dari pengujian di atas drive F: terdeteksi sebagai volume terenkripsi.

4.2 Pengujian Performansi Disk Terenkripsi

Pada pengukuran performa ini menggunakan *tools* CrystalDiskMark 6.0.2 x64. Yang diuji adalah perbandingan kecepatan baca dan tulis dengan dan tanpa enkripsi pada beberapa ukuran file dan mode

	Kecepatan Rata-rata (MB/s)		
	Tanpa Enkripsi	Dengan Enkripsi	Perbedaan
Read	23,83	23,88	0,21%
Write	8,623	10,09	16,93%

baca-tulis disk Seq Q32T1, yaitu mode *Sequential (Block Size=128KiB) Read/Write with multi Queues & Threads.*

1. Parameter 1

Spesifikasi :
Ukuran File : 50MB

Tabel 3 Parameter 1

2. Parameter 2

Spesifikasi :
Ukuran File : 500MB

Tabel 4 Parameter 2

3. Parameter 3

Spesifikasi :
Ukuran File : 1GB

Tabel 5 Parameter 3

	Kecepatan Rata-rata (MB/s)		
	Tanpa Enkripsi	Dengan Enkripsi	Perbedaan
Read	23,59	24,3	3,0 %
Write	9,384	9,883	5,32%

V. KESIMPULAN

Dari perancangan dan implementasi serta pengujian VeraCrypt dapat disimpulkan sebagai berikut:

1. VeraCrypt sebagai *tools* enkripsi file dan disk dapat mengenkripsi disk sehingga tidak bisa dibaca tanpa aplikasi VeraCrypt dengan key yang tepat
2. Dalam percobaan pengujian spesifikasi server, diperoleh hasil pengujian uji baca tulis dengan perbedaan maksimal 16,93% yang diperoleh saat pengetesan write ukuran file 50MB.

Dari hasil pengujian di atas dapat disimpulkan bahwa VeraCrypt berhasil mengamankan data sehingga tidak dapat dibuka oleh yang tidak berhak dan data tetap dapat dibaca dan ditulis tanpa ada perbedaan signifikan dengan disk yang tidak dienkripsi.

DAFTAR PUSTAKA

Padate, Roshni dkk. May 2014 "Encryption and Decryption of Text using AES Algorithm" International Journal of Emerging Technology and Advanced Engineering. ISSN 2250-2459

Putra, Soni Harza, dkk., 2013 "Implementasi Algoritma Kriptografi ADVANCED ENCRYPTION STANDARD (AES) Pada Kompresi Data Teks". Jurnal Teknologi Informasi, Universitas Brawijaya Malang. 2013.

Munir, Rinaldi. Kriptografi. 2006 Bandung : Penerbit Informatika.

R.Kristoforus JB, dkk. 2012 "Implementasi Algoritma Rijndael Untuk Enkripsi dan Dekripsi Pada Citra Digital".Seminar Nasional Aplikasi Teknologi Informasi 2012 (SNATI 2012). ISSN: 1907-5022. 2012

Voni Yuniati, dkk. April 2009 "Enkripsi dan Dekripsi Dengan Algoritma AES-256 Untuk Semua Jenis File". Jurnal Informatika Volume 5 No. 1 April 2009

Ellis, Will. June 2019" VeraCrypt Review: A Superb Open-Source Disk Encryption Software". PrivacyAustralia.net https://-privacyaustralia.net/veracrypt-review/ (diakses pada 6 Juni 2019, 8:32 WIB)

Anonim. 2018."VeraCrypt Introduction" https://www.veracrypt.fr/en/Introduction.html(diakses pada 17 Mei 2019, 10:37 WIB).

Anonim. 2018." AES-XTS Block Cipher Mode is used in Kingston's Encrypted USB Flash Drives" https://www.kingston.com/us/solutions/data-security/xts-encryption (diakses pada 25 Mei 2019, 21:08 WIB)