JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

IMPLEMENTASI ALGORITMA *RIJNDAEL* DALAM ENKRIPSI DAN DEKRIPSI GAMBAR DIGITAL BERBASIS WEB

Ade Chandra Saputra [a], Agus Sehatman Saragih [b]
[a] Universitas Palangka Raya, Jl.H.Timang Palangka Raya
[b] Universitas Palangka Raya, Jl.H.Timang Palangka Raya
[1] adechandra@it.upr.ac.id; [2] asaragih@gmail.com
* corresponding author

ARTICLE INFO

Keywords
Rijndael algorithm
Image Security System.

ABSTRACT (10PT)

More and more abuse of digital images, data or information that is personal in nature can be easily known by others who are not entitled through digital images. This can cause material and immaterial losses to people whose personal information is misused by others. Then the application applies the Rijndael algorithm to secure digital image images which contain information or data that is personal in nature. In securing digital images, the Rijndael algorithm is used to protect the information contained in these images, this algorithm runs with processes such as SubBytes, ShiftRows, MixColumns, and AddRoundKey. The methodology applied is data collection methods such as field studies and literature studies, then methods of developing Waterfall software (Communication, Planning, Modeling, Construction, and Deployment) for system design. The results of the test analysis get an accuracy value of 100% from the 14 image files tested, all of them were successfully encrypted and decrypted so that it returned to the original form of the original image. For further development, this application can input the data files of other documents and increase the key length to 196 bits and 256 bits.

1. Pendahuluan

Keamanan adalah keadaan bebas dari bahaya. Keamanan dapat diimplementasikan kedalam berbagai hal, termasuk data dan informasi. Data merupakan salah satu aset penting dalam kelangsungan hidup perusahaan, instansi – instansi pemerintahan, institusi – institusi pendidikan bahkan untuk pribadi. Semua orang memiliki sesuatu yang ingin keamanannya terjaga dan tidak terpublikasi dengan mudah, seperti uang, surat-surat berharga, barang-barang berharga dan lain-lain. Tidak hanya berhenti di situ saja yang membutuhkan keamanan, namun data atau informasi yang dimiliki, membutuhkan keamanan yang memadai, karena data atau informasi merupakan salah satu aset yang penting, yang jika tidak dijaga dengan baik dapat menimbulkan kerugian materil atau immateril. Data atau informasi tersebut dapat berupa teks, citra/gambar, audio , dan video. Dengan adanya kemajuan teknologi, data atau informasi dapat disajikan dalam bentuk digital. Akan tetapi, bentuk penyimpanan seperti ini sangat rentan aspek keamanannya. Data dapat dengan mudah diganti, dimanipulasi, dihilangkan, atau bahkan disalahgunakan. Salah satu data yang sering dimanipulasi dan disalahgunakan adalah gambar digital. Gambar yang dimanipulasi dapat merusak nama baik seseorang. Hal ini tentu saja melanggar hak privasi setiap orang. Dengan demikian usaha dalam mengamankan data digital menjadi hal yang sangat penting dan sangat mendesak. Adapun masalah keamanan dalam suatu data atau informasi dapat diatasi salah satunya dengan menerapkan kriptografi, yaitu ilmu atau seni yang menggunakan matematika untuk mengamankan suatu informasi. Pengamanan ini dilakukan dengan menjalankan algoritma enkripsi (mengubah informasi awal menjadi informasi baru yang disamarkan dengan menggunakan suatu kunci) dan dekripsi (mengubah kembali menjadi informasi awal).

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

Hampir setiap orang menjadikan gambar digital sebagai sarana untuk menyimpan informasi penting, misalnya bukti-bukti transaksi, surat-surat berharga, foto-foto yang berisikan informasi pribadi yang tidak ingin dipublikasikan dan lain-lain. Hal inilah yang menyebabkan perlindungan gambar digital menjadi sangat penting.

Permasalahan yang akan dibahas pada jurnal ini adalah sebagai berikut :

1. Bagaimana implementasi algoritma *Rijndael* dalam enkripsi gambar digital sehingga gambar digital tidak dapat dibaca atau dimengerti oleh pihak lain yang tidak berhak ?
2. Bagaimana hasil dekripsi dapat mengembalikan file hasil enkripsi ke bentuk awal ?

Berdasarkan rumusan masalah di atas, maka tujuannya adalah untuk mengimplementasikan Algoritma *Rijndael* yang dapat digunakan untuk mengenkripsi dan dekripsi gambar digital. Sehingga gambar digital tidak dapat dilihat dan dibaca oleh orang yang tidak berhak atau tidak memiliki kata kunci untuk dekripsi. Hal ini guna mengamankan hak-hak dan privasi pribadi terhadap informasi yang terkandung didalamnya.

2. Metodologi Penelitian

Jelaskan metode preparasi dan teknik karakterisasi yang digunakan. Jelaskan dengan ringkas, tetapi tetap akurat seperti ukuran, volume, replikasi dan teknik pengerjaan. Untuk metode baru harus dijelaskan secara rinci agar peneliti lain dapat mereproduksi percobaan. Sedangkan metode yang sudah mapan bisa dijelaskan dengan memetik rujukan.

2.1. Metode Pengembangan Sistem

Dalam tahap perancangan sistem ini menggunakan metode *Waterfall* (Roger S. Pressman, 2010) dengan tahap-tahap sebagai berikut :

1. *Communication* (Komunikasi)

   Pada tahap ini merupakan tahap analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data yang diperlukan. Hasil pada tahap komunikasi yaitu data-data yang dibutuhkan dalam pembuatan sistem.
2. *Planning* (Perencanaan)

   Proses perencanaan merupakan lanjutan dari proses *communication* (komunikasi). Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software,* termasuk rencana yang akan dilakukan.
3. *Modeling* (Pemodelan)

   Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding.* Proses ini terbagi menjadi 2, yaitu analisis dan desain. Pada analisis menggunakan *Data Flow Diagram* (DFD) yang akan menghasillkan *context diagram*.. Pada desain akan mendesain tabel, desain navigasi menggunakan *Sitemap* dan desain antarmuka/representasi *interface.*
4. *Construction* (Konstruksi)

   Construction merupakan proses membuat kode (*coding*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user.* Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software,* artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki. Untuk menerjemahkan kode-kode program dengan menggunakan bahasa pemrograman *PHP* dan pengujian dilakukan menggunakan *Blackbox Testing.*
5. *Deployment* (Penyerahan)

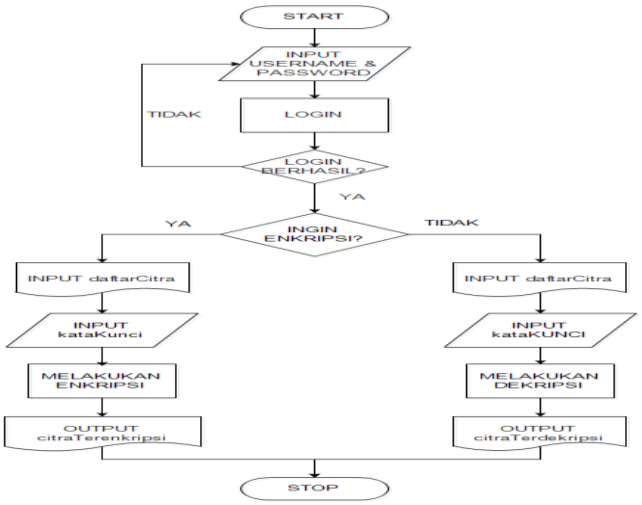
   Tahapan ini merupakan tahap akhir dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus evaluasi jika ada kekurangan dan dilakukan pemeliharaan secara berkala.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

2.2. Arsitektur Sistem

Arsitektur Sistem Adapun tahapan dalam enkripsi dan dekripsi gambar digital adalah sebagai berikut :

a. Langkah pertama yang dilakukan adalah program akan menampilkan halaman utama sebagai menu yang memberikan pilihan kepada pengguna untuk memilih enkripsi atau dekripsi.
b. Langkah kedua adalah memilih fitur yang ingin dijalankan baik itu enkripsi maupun dekripsi.
c. Langkah ketiga, pengguna akan diminta memasukan file citra gambar untuk enkripsi atau file citra terenkripsi untuk dekripsi.
d. Langkah keempat, pengguna diminta memasukan kunci dengan Panjang maksimal 16 karakter.
e. Langkah kelima, pengguna melakukan proses enkripsi dan dekripsi sebagaimana fitur yang dipilih.
f. Program menampilkan rincian dan mengeluarkan output file-file yang telah diolah berdasarkan fitur.

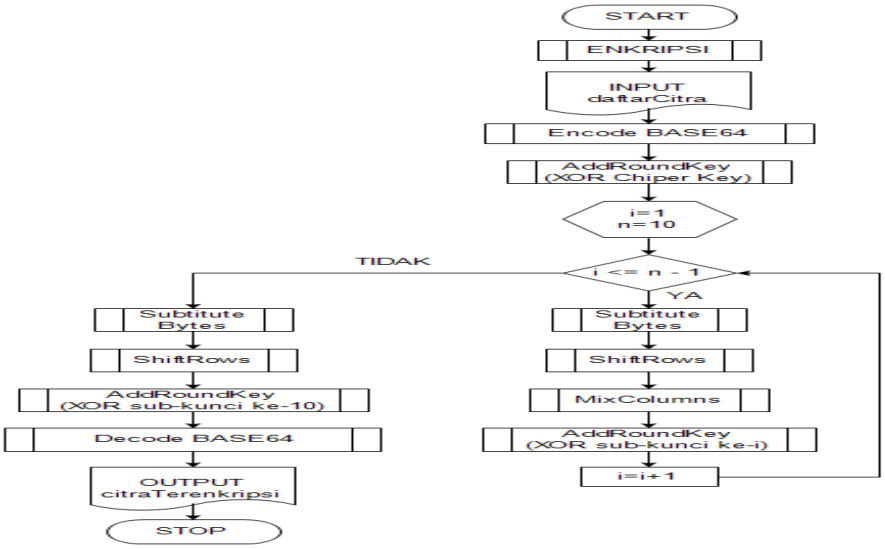


Gambar 1 Arsitektur Sistem

Gambar 1 menjelaskan cara kerja aplikasi dari awal sampai berakhir, dimulai dari kegiatan login, input file gambar dan kunci, kemudian memilih fitur enkripsi atau dekripsi yang akan dilakukan kemudian aplikasi akan mengeluarkan output berupa file gambar digital.

Gambar 2 menjelaskan cara kerja algoritma rijndael dalam mengenkripsi file gambar digital. AddRoundKey pada dasarnya adalah mengkombinasikan chiper teks yang sudah ada dengan roundkey dengan hubungan XOR. Kemudian untuk enkripsi 128 bit memerlukan iterasi sebanyak 10 kali, iterasi ini berisi proses subbytes yaitu menukar ini matriks dengan tabel pada rijndael S-Box, shiftrows yaitu sebuah proses dengan melakukan pergeseran pada tiap baris, mixcolumns yaitu mengalikan tiap elemen dari blok chiper dengan matriks, AddRoundKey.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]



Gambar 2 Arsitektur Enkripsi Rijndael

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3 Tabel S-Box

Gambar 3 merupakan tabel S-Box yang digunakan saat melakukan subBytes. Elemen dari matriks awal akan berubah nilainya karena dikombinasikan atau dicocokan nilainya dengan nilai baru yang ada pada tabel S-box.

Gambar 4 Arsitektur Dekripsi Rijndael

Gambar di 4 adalah alur kerja dekripsi, proses yang dilakukan pada dekripsi merupakan kebalikan dari proses enkripsi. XOR dengan subkunci ke 10 merupakan hal pertama kali yang dilakukan kemudian melakukan pergeseran baris secara terbalik dan menukar isi matriks dengan S-Box (gambar 4). kemudian melakukan iterasi sebanyak 10 kali dengan addRoundKey, MixColumns, ShiftRows,

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

dan SubBytes secara terbalik. Dan yang terakhir melakukan XOR pada chiperkey sehingga sistem menghasilkan output berupa gambar digital seperti aslinya.

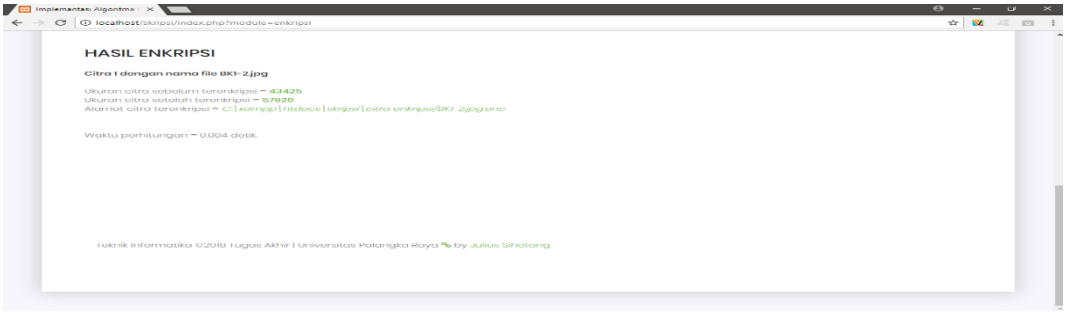
3. Hasil dan Pembahasan

Pembahasan terhadap hasil implemantasi yang dirancang terdapat dua jenis pengujian yang dilakukan adalah :

1. Pengujian hasil dekripsi apakah sama dengan *file* awal yang di-*input*.

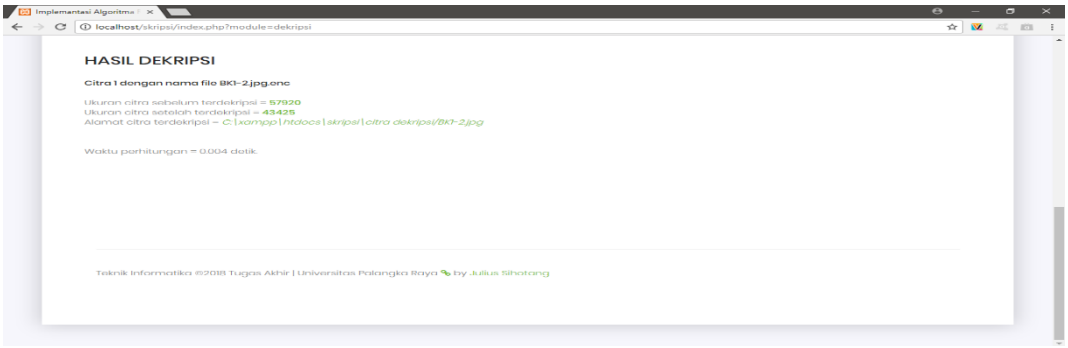
Gambar 1 File Input citra RGB dengan format JPG

Program diberikan inputan berupa *file* citra berwarna dengan format JPG dan dengan menggunakan kunci “teskunci” sehingga menghasilkan *file* terenkripsi seperti gambar 5, dimana ukuran citra gambar sebelum dienkripsi adalah 43,4 Kb dan setelah dienkripsi menjadi 57,9 Kb, serta durasi proses enkripsi adalah 0,004 detik.



Gambar 2 Hasil Enkripsi File Input Citra RGB dengan format JPG

Program kemudian diberikan *input*-an berupa *file* yang telah dienkripsi sehingga menghasilkan seperti gambar 6. Program menampilkan ukuran *file* terenkripsi sebesar 57,9 Kb, ukuran *file* setelah didekripsi sebesar 43,4 Kb, dan durasi proses dekripsi adalah 0,004 detik.



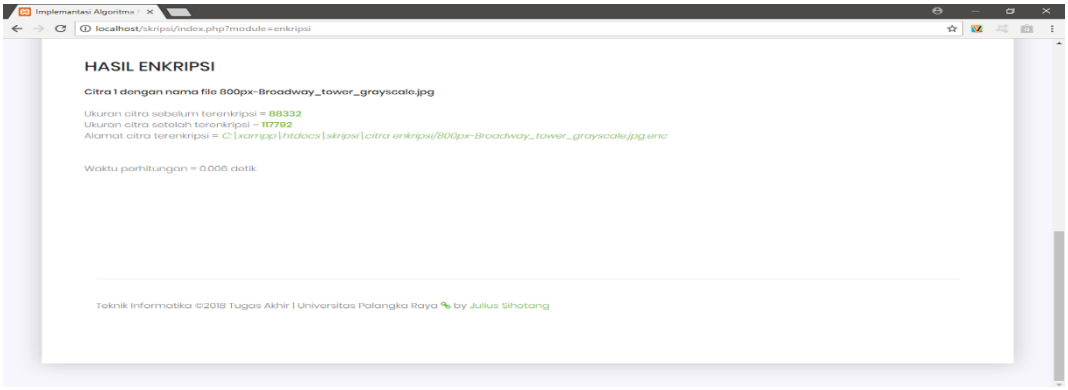
Gambar 3 Program Melakukan Dekripsi File Input Citra Terenkripsi

Gambar 7 menampilkan citra yang telah didekripsi memiliki kesamaan seperti citra sebelum dienkripsi, sehingga dapat dipastikan program berjalan dengan baik pada citra RGB berformat JPG.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

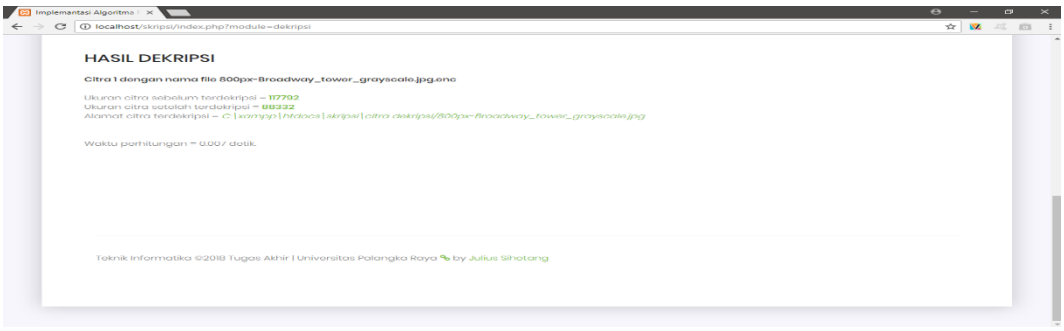
Gambar 4 File Input citra Grayscale dengan format JPG

Gambar 8 menampilkan *file* citra *grayscale* yang akan dienkripsi menggunakan program dengan kunci "teskunci". Program melakukan eksekusi fungsi enkripsi pada *file* citra *grayscale*, program menampilkan hasil seperti gambar 8, dimana ukuran citra gambar sebelum dienkripsi adalah 88,3 Kb dan setelah dienkripsi menjadi 117,8 Kb, serta durasi proses enkripsi adalah 0,006 detik.



Gambar 5 Program Melakukan Enkripsi Pada Citra Grayscale

Program kemudian diminta untuk melakukan dekripsi pada *file* citra terenkripsi sebelumnya. Program menampilkan hasil proses dekripsi yang berisi informasi seperti gambar 4.14 yaitu ukuran citra sebelum terdekripsi sebesar 117,8 Kb, ukuran setelah terdekripsi sebesar 88,3 Kb, dan mengalami proses selama 0,07 detik.



Gambar 6 Program Melakukan Dekripsi Pada Citra Grayscale

Hasil dari dekripsi *file* citra terenkripsi kemudian dibuka menggunakan aplikasi penampil gambar dan menampilkan gambar sesuai dengan gambar sebelum dienkripsi.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

Gambar 7 Hasil Dekripsi File Citra Grayscale Terenkripsi

2. Durasi enkripsi dan dekripsi jenis *file* citra (RGB atau *Grayscale*) dengan ukuran dan kunci yang sama. Kedua gambar sebelumnya dilakukan *resize* menjadi 500 x 500 *pixel* dengan aspek rasio 1:1 sehingga mendapatkan ukuran lebar dan tinggi yang sama untuk menghasilkan pengukuran waktu proses yang lebih efektif.

Tabel 1 Pengujian Durasi Enkripsi Terhadap Jenis Citra

Citra	Ukuran Citra		Durasi (Detik)
	Sebelum Enkripsi (Bytes)	Sesudah Enkripsi (Bytes)	
	105415	140576	0,008
	162233	216336	0,010

Tabel 2 Pengujian Durasi Dekripsi Terhadap Jenis Citra

Citra	Ukuran Citra		Durasi (Detik)
	Sebelum Dekripsi (Bytes)	Sesudah Dekripsi (Bytes)	
	140576	105415	0,010

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

	216336	162233	0,017

Dari hasil pengujian diatas didapatkan hasil bahwa citra *grayscale* dan RGB dengan resolusi yang sama yaitu 500x500px memiliki ukuran citra yang berbeda, dalam hal ini citra RGB lebih besar dibandingkan citra *grayscale* karena informasi yang tersimpan didalam citra RGB lebih banyak dibanding dengan citra *grayscale*, sehingga dalam pengujian ini citra *grayscale* lebih cepat diproses dibanding dengan citra RGB baik dalam hal enkripsi maupun dekripsi.

3. Pengaruh perbedaan kunci pada hasil enkripsi dan dekripsi. Pada pengujian ini dilakukan *input* data pada *file* citra yang sama dengan kunci yang berbeda baik dari panjang kunci dan jenis kunci.

Gambar 12 merupakan gambar yang dipakai dalam pengujian ini, gambar ini memiliki dimensi sebesar 275x149 *pixels*.

Gambar 8 Citra Gambar Input Pengujian Beda Kunci

Tabel 3 Pengaruh Perbedaan Kunci Pada Enkripsi

Jenis Kunci	Kunci	Ukuran Sebelum (Bytes)	Ukuran Sesudah (Bytes)	Durasi (Detik)
Huruf saja (16 Karakter)	abcdefghijklmnop	50606	67504	0,007
Angka saja (16 Karakter)	1234567891234567	50606	67504	0,007
Simbol saja (16 Karakter)	<>,.-=+_&*%#^&!~	50606	67504	0,005
Huruf, angka dan simbol (16 Karakter)	teskunci1234=+->	50606	67504	0,007
Huruf saja (x<16 Karakter)	tes	50606	67504	0,007
Angka saja (x<16 Karakter)	1495	50606	67504	0,005
Simbol saja (x<16 Karakter)	+=>[&	50606	67504	0,006
Huruf, angka dan simbol (x<16 Karakter)	tes123&=	50606	67504	0,005

Dapat dilihat pada tabel 3 pengujian perbedaan jenis kunci dan panjang kunci tidak berpengaruh pada ukuran hasil enkripsi, hal ini dikarenakan proses enkripsi yang dijalankan merupakan enkripsi 128 bit yang membutuhkan kunci 128 bit juga.

Code : $key = str_repeat ($key , floor(16 / strlen($key))+1);

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

Kode diatas merupakan kode untuk melakukan pengulangan kunci, kode ini dijalankan apabila kunci yang di-*input* panjangnya kurang dari 128 *bit* atau 16 *byte*. Terdapat operator *division* yang digunakan untuk mengetahui nilai *div* dari 16 dan panjang kunci. Hasil kemudian ditambahkan dengan satu serta dibulatkan dengan fungsi *floor*, sehingga kita mengetahui berapa kali kunci harus diulang agar kunci yang di-*input* memiliki panjang lebih dari atau sama dengan 16.

Code : $key = substr($key,0,16);

Kode diatas merupakan kode yang dijalankan setelah pengulangan kunci dilakukan untuk memastikan panjang kunci yang di-*input* adalah 128 bit. Pengaruh perbedaan jenis kunci dan panjang kunci pada enkripsi terhadap ukuran dan durasi proses enkripsi berlaku sama dengan proses dekripsi, hal ini dikarenakan pada fungsi dekripsi dijalankan fungsi *str_repeat* dan *substr* juga.

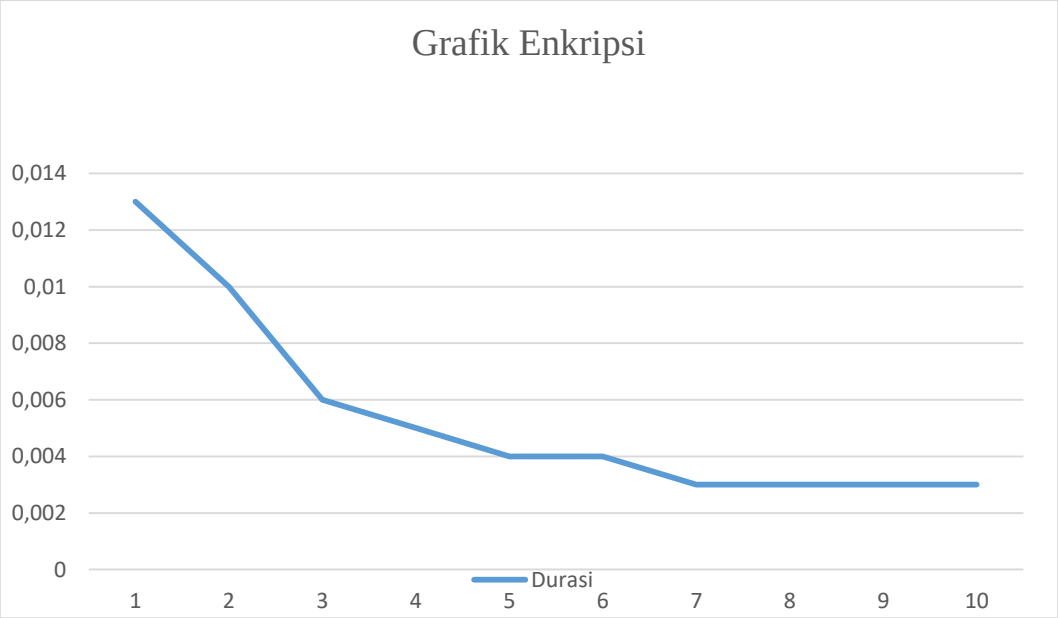
4. Durasi enkripsi dan dekripsi terhadap perbedaan resolusi citra *input*. Pada pengujian ini akan diambil 10 sampel dari gambar yang sama tetapi dengan resolusi yang berbeda. Semuanya akan dienkripsi dan dekripsi dengan kunci yang sama yaitu “teskunci”.

Gambar 9 Citra Uji Utama

Tabel *4* Uji Durasi Enkripsi Berdasarkan Ukuran

Data	Ukuran (pixel)	Ukuran Sebelum Enkripsi (Bytes)	Ukuran Sesudah Enkripsi (Bytes)	Durasi (Detik)
1	650x350	149848	199824	0,013
2	585x315	119230	158992	0,010
3	527x284	104213	138976	0,006
4	422x228	80877	107856	0,005
5	338x183	62969	83984	0,004
6	275x149	50606	67504	0,004
7	224x112	41638	55536	0,003
8	180x98	35545	47424	0,003
9	144x79	30368	40512	0,003
10	116x64	26083	34800	0,003

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]

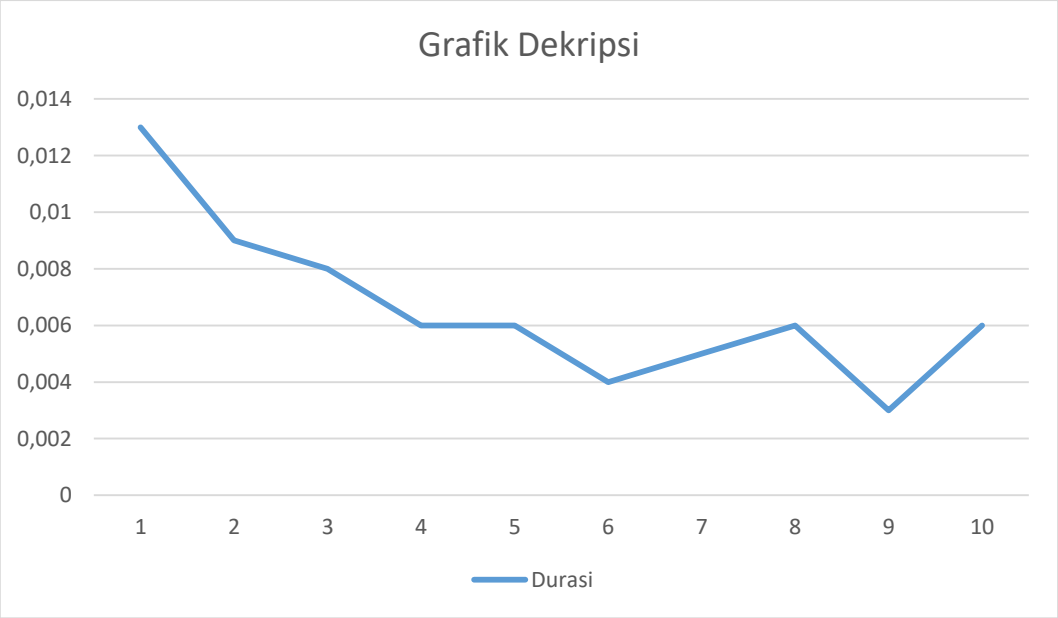


Gambar 10 Grafik Enkripsi

Tabel 5 Uji Durasi Dekripsi Berdasarkan Ukuran

Data	Ukuran (pixel)	Ukuran Sebelum Dekripsi (Bytes)	Ukuran Sesudah Dekripsi (Bytes)	Durasi (Detik)
1	650x350	199824	149848	0,013
2	585x315	158992	119230	0,009
3	527x284	138976	104213	0,008
4	422x228	107856	80877	0,006
5	338x183	83984	62969	0,006
6	275x149	67504	50606	0,004
7	224x112	55536	41638	0,005
8	180x98	47424	35545	0,006
9	144x79	40512	30368	0,003
10	116x64	34800	26083	0,006

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 1]
[Januari 2020]



Gambar 11 Grafik Enkripsi

Dari tabel 4 dan 5 menunjukan bahwa terdapat perbedaan dalam *running time* dengan ukuran piksel yang berbeda. Makin besar ukuran piksel maka kumungkinan makin besar pula *running time* yang dibutuhkan. Dari semua uji coba enkripsi dan dekripsi yang dilakukan diperoleh nilai akurasi program sebagai berikut :

$$nilai\ akurasi = \frac{File\ citra\ yang\ berhasil}{Seluruh\ file\ citra\ yang\ diuji}\ x\ 100\%$$

$$nilai\ akurasi = \frac{14}{14}\ x\ 100\%$$

$$nilai\ akurasi = 100\%$$

4. Kesimpulan

Berdasarkan hasil penelitian , maka dapat ditarik kesimpulan sebagai berikut. Program ini melakukan beberapa proses dalam menjalankan fungsinya yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Tahapan-tahapan tersebut dijalankan sebanyak 10 round, karena kunci yang dipakai adalah 128 bit. Implementasi algoritma Rijndael dalam enkripsi dan dekripsi citra gambar digital, dapat ditarik kesimpulan sebagai berikut:

1. Hasil enkripsi merupakan sekumpulan kombinasi karakter yang tidak sama seperti aslinya, sehingga tidak dapat dimengerti oleh manusia biasa atau pihak – pihak yang tidak mengetahui kunci dan cara kerja algoritma yang dipakai dalam enkripsi tersebut.
2. Hasil enkripsi selalu sama dengan hasil dekripsi walaupun gambar yang di-input berbeda jenis, seperti citra RGB dan Grayscale. Keduanya akan kembali seperti keadaan awal sebelum dienkripsi.
3. Panjang kunci yang diinputkan tidak banyak mempengaruhi durasi enkripsi maupun dekripsi, karena basis kunci akan selalu menjadi 16 bytes meskipun kunci yang di-input-kan kurang dari 16 bytes.
4. Ukuran file yang dilakukan enkripsi dan dekripsi sangat berpengaruh dalam runnning time, semakin besar ukuran file yang diproses maka waktu yang dibutuhkan program untuk menyelesaikannya semakin lama juga. Hal ini berlaku sebaliknya, semakin kecil program maka waktu yang diperlukan akan semakin kecil.

5. Nilai akurasi program adalah 100% , hal ini didapat dari 14 data file citra gambar digital yang digunakan sebagai alat uji dan semuanya berhasil dienkripsi dan dikembalikan kebentuk awal dengan dekripsi.

Daftar Pustaka

[1] Aprianto dkk,. 2014, *Rancang Bangun Aplikasi Enkripsi dan Dekripsi Gambar digital Menggunakan Algoritma Rijndael Berbasis Java SE.* STMIK GI MDP.
[2] Bendi, 2012, *Implementasi Algoritma Rijndael Untuk Enkripsi dan Dekripsi Pada Citra Digital.* Universitas Katolik Musi Charitas.
[3] Bruce Schneier, 1996, *Section 14.1 GOST, in Applied Cryptography, Second Edition.* ISBN 0-471-11709-9
[4] Deni Darmawan & Deden Hendra Permana. 2013. *Desain dan Pemrograman*
[5] *Website.* Penerbit PT Remaja Rosdakarya : Bandung.
[6] Dony Ariyus, 2006, *Kriptografi : Keamanan Data dan Komunikasi*, Cetakan Pertama, Penerbit GRAHA ILMU, Yogyakarta.
[7] Dony Ariyus, 2008, *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi*, Penerbit GRAHA ILMU, Yogyakarta.
[8] Irfan, 2016, *Aplikasi Enkripsi Citra Menggunakan Algoritma Kriptografi Arnold Cat Map Dan Logistic Map*. STMIK Bumigora : Mataram.
[9] Jogiyanto Hartono, 2005, *Analisis dan Desain Sistem Informasi. Pendekatan Tersetruktur Teori dan Praktis Aplikasi Bisnis*, Andi: Yogyakarta.
[10] Kustiyaningsih, Yeni. 2011, *Pemrograman Basis Data Berbasis Web menggunakan PHP dan MySQL*. Penerbit Graha Ilmu: Yogyakarta.
[11] Munir, 2012, *Analisis Keamanan Algoritma Enkripsi Citra Digital Menggunakan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif.* ITB: Bandung.
[12] Setyanigsih Emy, 2015, *Kriptografi & Implementasinya Menggunakan Matlab*, Andi Publisher: Yogyakarta.
[13] Soumya, 2013. *Design and Implementation of Rijndael Encryption Algorithm Based on FPGA*. JITS: India.
[14] Supriyanto, 2008, *Teknik Informasi & Komunikasi SMP Kelas VII*, Yudhistira : Yogyakarta.
[15] Suryadi dkk, 2014. *Implementasi Algoritma Enkripsi Citra Digital Menggunakan Skema Transposisi Berbaris Pada Fungsi Chaos.* FMIPA Universitas Indonesia.
[16] Rifki Sadikin, 2012, *Kriptografi Untuk Keamanan Jaringan*. Penerbit ANDI: Yogyakarta.
[17] Rinaldi Munir, 2006, Kriptografi, Penerbit INFORMATIKA, Bandung.