

# RANCANG BANGUN WEBSITE BADAN PENGAWAS PEMILIHAN UMUM (BAWASLU) KALIMANTAN TENGAH

Ade Chandra Saputra<sup>a,1,\*</sup>, Agus Sehatman Saragih<sup>b,2</sup>

<sup>a</sup>Jurusan Teknik Informatika FT UPR, Jl H. Timang Tunjung Nyaho

<sup>b</sup>Jurusan Teknik Informatika FT UPR, Jl H. Timang Tunjung Nyaho

<sup>1</sup>adechandra@it.upr.ac.id \*; <sup>2</sup>assaragih@it.upr.ac.id

\* corresponding author

## ARTICLE INFO

## ABSTRACT (10PT)

### Keywords

Algoritma Rijndael

Waterfall

digital image

*The misuse of digital image images is increasing, data or information that is of a personal nature can easily be 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 of a personal nature. In securing digital image images, Rijndael's algorithm is used to protect the information contained in the image, the 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 Waterfall software development methods (Communication, Planning, Modeling, Construction, and Deployment) for system design.*

## 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).

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 adalah sebagai berikut :

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

Berdasarkan permasalahan tersebut maka tujuan untuk mengimplementasikan algoritma *Rijndael* adalah 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

#### Kajian literatur dan pengembangan hipotesis

Dalam melakukan pengamanan data gambar digital diperlukan suatu cara yang harus dilakukan untuk bisa membuat gambar digital tidak bisa disalah gunakan oleh orang tertentu yang dapat merugikan pribadi ataupun kelompok dengan tidak mengubah integritas data gambar digital tersebut. Penggunaan aplikasi dalam melakukan pengenkripsian dan pendekripsioan menjadi salah satu cara yang efektif untuk melakukan pengamanan gambar digital. Aplikasi penerapan algoritma *Rijndael* sudah ada dibuat sebelumnya, tetapi basis dan tujuan aplikasi digunakan berbeda – beda. Pada penelitian berjudul “Rancang Bangun Aplikasi Enkripsi dan Dekripsi Gambar digital Menggunakan Algoritma *Rijndael* Berbasis Java SE” (Aprianto dkk, 2014). Penelitian ini menjelaskan Algoritma *Rijndael* dapat diimplementasikan atau diterapkan dalam teknik enkripsi dan dekripsi pada gambar digital berbasis Java SE. Variasi jenis kunci input (Angka saja, huruf saja, simbol saja, dan gabungan ketiganya) yang digunakan untuk proses enkripsi dan dekripsi tidak mempengaruhi ukuran ukuran piksel dan ukuran memori gambar digital asli atau file tidak mengalami perubahan, sehingga integritas citra terpenuhi.

Serta pada penelitian yang berjudul “Design and Implementation of *Rijndael* Encryption Algorithm Based on FPGA” (Soumya, 2013). Penelitian ini menjelaskan bahwa Algoritma *Rijndael* dapat diimplementasikan atau diterapkan bukan hanya untuk mengamankan data informasi perusahaan atau badan instansi besar tetapi juga dalam kehidupan sehari-hari. Mode pengenkripsian pada perancangan di makalah ini telah diubah untuk mengurangi ukuran chip (kunci enkripsi). Misalnya untuk kunci 128 bit, komputer akan membagi atau memisahkan setiap 32 bit untuk dilakukan pemrosesan enkripsi.

Pada penelitian “Implementasi Algoritma *Rijndael* Untuk Enkripsi dan Dekripsi Pada Citra Digital” (Bendi, 2012). Penelitian ini menjelaskan enkripsi dengan mode *chiperblock* ECB membuat citra asli yang memiliki banyak warna tunggal akan memiliki hasil enkripsi cenderung masih memiliki kesamaan yang kasat mata.

Pada penelitian yang berjudul “Implementasi Algoritma Enkripsi Citra Digital Menggunakan Skema Transposisi Berbaris Pada Fungsi Chaos” (Suryadi dkk, 2014). Penelitian ini menjelaskan bahwa fungsi chaos sangat cocok untuk merancang sarana untuk menunjukkan posisi baru dari pixel XY yang berasal dari citra dengan ukuran N x N pixel. Jadi setiap pixel pada citra diacak sedemikian rupa sehingga membuat citra baru yang sulit untuk dibaca oleh orang lain.

Pada penelitian yang berjudul “Aplikasi Enkripsi Citra Menggunakan Algoritma Kriptografi Arnold Cat Map Dan Logistic Map” (Irfan, 2016). Penelitian ini menjelaskan bahwa

ACM melakukan pengacakan pixel pada area gambar yang ada, ACM pada dasarnya hanya dapat digunakan pada gambar dengan panjang dan lebar yang sama atau memiliki rasio 1:1.

Pada penelitian yang berjudul “Analisis Keamanan Algoritma Enkripsi Citra Digital Menggunakan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif” (Munir, 2012). Penelitian ini menjelaskan bahwa enkripsi dengan menggunakan ruang kunci yang cukup besar kemungkinan brute force pada gambar digital hasil enkripsi memakan waktu yang sangat lama, hal ini membuat keamanan data hasil enkripsi tergolong sangat aman.

Sedangkan dalam penelitian ini pengenkripsian gambar digital dilakukan menggunakan kunci 128 bit, untuk melakukan enkripsi dengan panjang kunci 128 bit dibutuhkan 10 kali proses SubBytes (substitusi byte dengan menggunakan S-Box), ShiftRows (pergeseran baris-baris array state secara wrapping), MixColumns (mengacak data dimasing-masing kolom array state), dan AddRoundKey (melakukan XOR antara state sekarang dengan roundkey). Rijndael bekerja pada matriks berukuran 4 x 4 dimana tiap-tiap sel matriks terdiri dari 1 byte (8 bit). Pemrograman dilakukan dengan bahasa pemrograman PHP, untuk pengujian penelitian ini sendiri akan dilakukan membandingkan Runningtime program, ukuran gambar asli dan hasil enkripsi, dan banyaknya pixel yang berubah.

## 2. Metode Penelitian

Pada tahap ini, yang dilakukan adalah menganalisis kebutuhan-kebutuhan apa saja yang dibutuhkan. Metode referensi seperti pembelajari buku, jurnal dan informasi dari internet yang berhubungan dengan pembuatan aplikasi serta algoritma *Rijndael*. Melakukan pengamatan pada situs – situs sejenis untuk melengkapi fitur ini.

### 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 menghasilkan *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*.

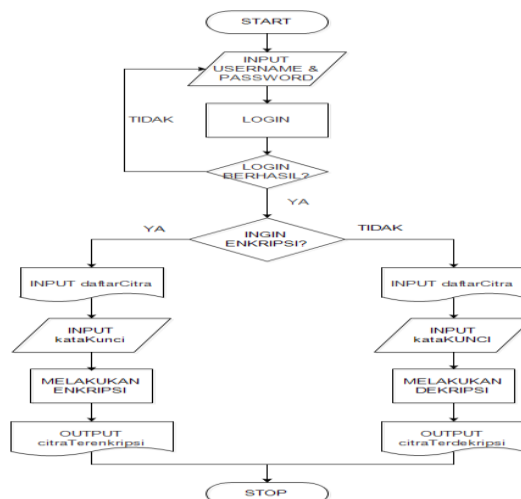
#### 5. *Deployment* (Penyerahan)

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.

#### 2.2. Arsitektur Sistem

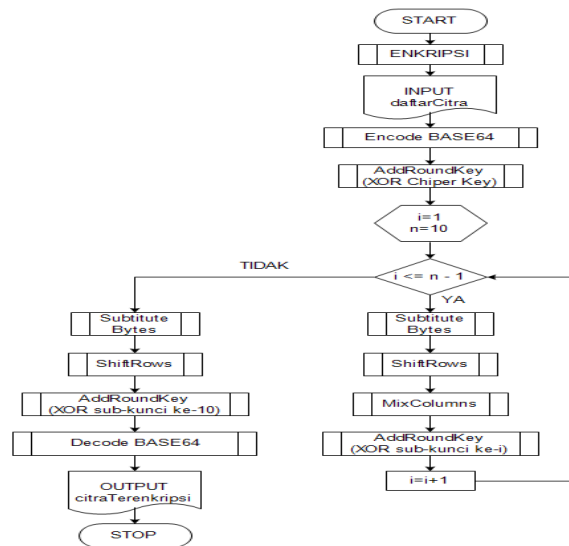
Adapun tahapan dalam enkripsi dan dekripsi gambar digital adalah sebagai berikut :

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



Gambar 3.1 Arsitektur Sistem

Gambar 3.1 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.



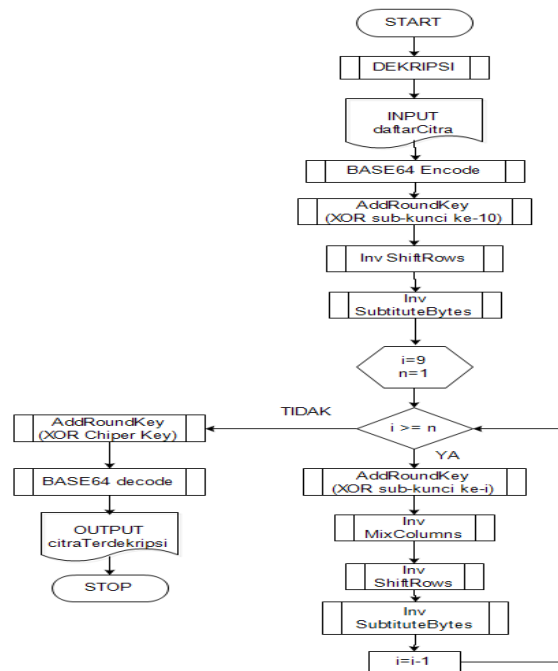
Gambar 3.2 Arsitektur Enkripsi Rijndael

Gambar 3.2 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*.

	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	d4	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 3.3 Tabel S-Box

Gambar 3.3 merupakan tabel *S-Box* yang digunakan saat melakukan *subBytes*. Elemen dari matriks awal akan berubah nilainya karena dikombinasikan atau dicocokkan nilainya dengan nilai baru yang ada pada tabel *S-box*.



Gambar 3.4 Arsitektur Dekripsi Rijndael

Gambar di 3.4 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 3.4). kemudian melakukan iterasi sebanyak 10 kali dengan *addRoundKey*, *MixColumns*, *ShiftRows*, dan *SubBytes* secara terbalik. Dan yang terakhir melakukan XOR pada *chiperkey* sehingga sistem menghasilkan output berupa gambar digital seperti aslinya.

### 3. Hasil Dan Pembahasan



Untuk membuktikan bahwa aplikasi yang telah dibuat dapat berjalan dan berfungsi dengan benar, maka pada tahap inilah aplikasi akan diimplementasikan atau skenario uji coba yang dapat menunjukkan bahwa hasil yang diperoleh dari uji coba tersebut sesuai dengan tujuan yang diharapkan. Berikut adalah proses implementasi dari program *Implementasi Algoritma Rijndael Dalam Enkripsi dan Dekripsi Citra Gambar Digital Berbasis Web*.

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

Program diberikan inputan berupa *file* citra berwarna dengan format JPG dan dengan menggunakan kunci “teskunci” sehingga menghasilkan *file* terenkripsi, 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.

Program kemudian diberikan *input*-an berupa *file* yang telah dienkripsi sehingga menghasilkan gambar. 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.

Tabel. 3.1. Durasi enkripsi dan dekripsi jenis *file* citra (RGB atau *Grayscale*) dengan ukuran dan kunci yang sama.

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

Tabel 3.2. Pengaruh perbedaan kunci pada hasil enkripsi dan dekripsi.

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

Tabel 3.3. Durasi enkripsi dan dekripsi terhadap perbedaan resolusi citra *input*.

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

Dari semua uji coba enkripsi dan dekripsi yang dilakukan diperoleh nilai akurasi program sebagai berikut :

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

$$\text{nilai akurasi} = \frac{14}{14} \times 100\%$$

$$\text{nilai akurasi} = 100\%$$

#### 4. KESIMPULAN

Kesimpulan yang dihasilkan dari penelitian ini

- 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.
- 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.
- 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*.
- Ukuran *file* yang dilakukan enkripsi dan dekripsi sangat berpengaruh dalam *running 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.
- 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.

Saran untuk penelitian yang akan datang adalah :

- Dapat menambah fitur untuk mengenkripsi *file* dengan format selain gambar digital.
- Dapat menambahkan kunci menjadi 196 dan 256 bit untuk melakukan enkripsi dan dekripsi.
- Dapat dikembangkan dengan menggunakan metode lain seperti *Blowfish*, *Twofish*, *Mars*, *RC6*, *Serpent* dan lain-lain.
- Dapat dikembangkan dengan bahasa pemrograman lain, contoh : Visual C++, C#, Java, dan Sebagainya

#### Daftar Pustaka

- Aprianto dk., 2014, *Rancang Bangun Aplikasi Enkripsi dan Dekripsi Gambar digital Menggunakan Algoritma Rijndael Berbasis Java SE*. STMIK GI MDP.
- Bendi, 2012, *Implementasi Algoritma Rijndael Untuk Enkripsi dan Dekripsi Pada Citra Digital*. Universitas Katolik Musi Charitas.
- Bruce Schneier, 1996, *Section 14.1 GOST*, in *Applied Cryptography, Second Edition*. ISBN 0-471-11709-9
- Deni Darmawan & Deden Hendra Permana. 2013. *Desain dan Pemrograman Website*. Penerbit PT Remaja Rosdakarya : Bandung.
- Dony Ariyus, 2006, *Kriptografi : Keamanan Data dan Komunikasi*, Cetakan Pertama, Penerbit GRAHA ILMU, Yogyakarta.
- Dony Ariyus, 2008, *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi*, Penerbit GRAHA ILMU, Yogyakarta.
- Irfan, 2016, *Aplikasi Enkripsi Citra Menggunakan Algoritma Kriptografi Arnold Cat Map Dan Logistic Map*. STMIK Bumigora : Mataram.
- Jogiyanto Hartono, 2005, *Analisis dan Desain Sistem Informasi. Pendekatan Terstruktur Teori dan Praktis Aplikasi Bisnis*, Andi : Yogyakarta.



- Kustiyaningsih, Yeni. 2011, *Pemrograman Basis Data Berbasis Web menggunakan PHP dan MySQL*. Penerbit Graha Ilmu : Yogyakarta.
- Munir, 2012, *Analisis Keamanan Algoritma Enkripsi Citra Digital Menggunakan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif*. ITB : Bandung.
- Setyaningsih Emy, 2015, *Kriptografi & Implementasinya Menggunakan Matlab*, Andi Publisher : Yogyakarta.
- Shiral, et. Al, 2015, *An Approach to Rijndael Algorithm*, tersedia pada <http://www.jiaats.com/Journals-Pdf/February-2015/jase/Feb-Jase-1.pdf> diakses pada tanggal 16 Februari 2018.
- Soumya, 2013. *Design and Implementation of Rijndael Encryption Algorithm Based on FPGA*. JITS : India.
- Supriyanto, 2008, *Teknik Informasi & Komunikasi SMP Kelas VII*, Yudhistira : Yogyakarta.
- Suryadi dkk, 2014. *Implementasi Algoritma Enkripsi Citra Digital Menggunakan Skema Transposisi Berbasis Pada Fungsi Chaos*. FMIPA Universitas Indonesia.
- Rifki Sadikin, 2012, *Kriptografi Untuk Keamanan Jaringan*. Penerbit ANDI : Yogyakarta.
- Rinaldi Munir, 2006, *Kriptografi*, Penerbit INFORMATIKA, Bandung.
- <http://searchsecurity.techtarget.com/definition/rijndael>, diakses pada tanggal 20 April 2017.
- <http://metode-algoritma.com/2013/06/algoritma-rijndael.html>, diakses pada tanggal 20 April 2017.
- <http://iliketowastemytime.com/2012/06/22/soviet-censorship-of-images-during-stalins-regime-t-pics>, diakses pada tanggal 21 April 2017.