Konversi Citra ke dalam Bentuk Teks Terenkripsi dengan Memanfaatkan Chiper Abjad Majemuk

Dadan Ramdan Mangunpraja [1)]

1) Jurusan Teknik Informatika, STEI ITB, Bandung, email: if14087@if.itb.ac.id

Abstract *– Konversi citra ke dalam bentuk teks pada dasarnya merupakan ide sederhana. Tujuannya untuk mengecoh para kriptanalis yang tidak akan menyangka kalau teks yang akan dianalisis untuk dipecahkan sebenarnya adalah citra. Sudah diketahui bahwa enkripsi citra cukup sulit dipecahkan bahkan untuk algoritma sederhana sekalipun. Hal ini dikarenakan sangat sulit untuk menganalisis distribusi dan keragaman informasi pixel-pixel pada citra. Citra yang dikonversi ke dalam bentuk teks yang berisi karakter-karakter alfabet akan lebih menyulitkan para kriptanalis lagi, karena mereka akan digiring ke dalam kesesatan dengan mengira bahwa pesan tersebut adalah pesan teks, bukan pesan citra.*

Kata Kunci*: Enkripsi Citra*

1. PENDAHULUAN

Media citra sudah banyak digunakan untuk menyampaikan pesan. Seperti halnya pesan teks dalam menjaga kerahasiaannya, pesan citra juga memerlukan teknik-teknik enkripsi yang sebisa mungkin sederhana tapi sukar dipecahkan. Hal yang seudah dilakukan dalam kriptografi citra adalah mengenkripsi citra ke dalam bentuk citra lagi dengan algoritma tertentu. Meskipun citra rahasia hasil enkripsi masih tergolong sulit untuk dipecahkan (bahkan untuk citra yang dienkripsi dengan algoritma sederhana), bukan tidak mungkin teknik-teknik pemecahan citra terenkripsi akan terus dikembangkan, sehingga tingkat keamanan metoda enkripsi citra ke dalam bentuk citra lagi akan berkurang.

Ide untuk mengantisipasi hal tersebut adalah dengan melakukan konversi citra ke dalam bentuk teks kemudian teks tersebut dienkripsi menggunakan algoritma tertentu. Hal tersebut menjadikan kriptanalis lebih sulit memecahkan *chipertext* karena tidak tahu apakah *chipertext* tersebut adalah hasil enkripsi citra atau teks. Pada akhirnya teknik seperti ini yang bisa dilakukan dengan algoritma sederhana sudah memiliki tingkat kesulitan tinggi, maka jika menggunakan algoritma yang lebih kompleks tentu akan jauh lebih sulit lagi untuk dipecahkan.

Dalam kenyataannya, teknik seperti ini juga tidak hanya berlaku untuk pesan citra saja. Teknik ini juga bisa dilakukan untuk pesan teks dengan memanipulasi pesan teks sedemikian rupa sehingga seolah olah menjadi pesan gambar, misalnya dengan menggunakan fasilitas editior gambar (*photoshop, coreldraw, paint*, dan sebagainya). Dengan demikian pesan teks dapat dienkripsi dengan menggunakan teknik ini menjadi pesan teks lagi tapi dengan tingkat keulitan yang lebih tinggi untuk dipecahkan walaupun algoritmanya sederhana.

2. PENJELASAN ALGORITMA

2.1 Garis Besar Teknik

Penjelasan singkat teknik enkripsi citra ke dalam bentuk teks adalah sebagai berikut:

1. Tiap pixel pada citra terdapat informasi /nilai. Nilai-nilai tersebut dikonversi ke dalam karakter huruf seperti mengkonversi bilangan desimal ke dalam heksadesimal. Aturan yang digunakan pun sederhana yaitu mengganti nilai dengan abjad yang bersesuaian dengan urutan alfabet sebagai berikut:

Tabel 1: konversi nilai menjadi alphabet

Nilai	Konversi		Nilai	Konversi
0	**A**		8	**I**
1	**B**		9	**J**
2	**C**		10	**K**
3	**D**		11	**L**
4	**E**		12	**M**
5	**F**		13	**N**
6	**G**		14	**O**
7	**H**		15	**P**

   Misalkan untuk suatu *pixel* pada citra bertipe .jpg didapatkan nilai **R** = 145, **G** = 29, dan **B** = 12. Untuk **R** = 145, karakter pertamanya adalah 145 *div* 16 = 9, menjadi **J** setelah dikonversi. Karakter keduanya adalah 145 *modulo* 16 = 1, menjadi **B** setelah dikonversi. Hal yang sama dilakukan terhadap nilai G dan B sehingga untuk *pixel* tersebut akan didapatkan rangkaian huruf **JBBNAM.**
2. Karena konversi tersebut terbatas hanya sampai 16 alfabet, maka perlu dilakukan suatu cara agar karakter yang muncul menjadi lebih beragam dari A sampai Z. Cara yang mudah

dan sederhana adalah dengan menggunakan enkripsi *chipper* abjad majemuk menggunakan kunci tertentu pada tiap-tiap alfabet yang dihasilkan. Sebagai contoh untuk sebuah arsip gambar dengan tipe 24 bit (3 *byte*) untuk tiap *pixel*-nya setelah melalui proses 1 di atas didapatkan rangkaian huruf sebagai berikut:

Tabel 2: contoh hasil konversi proses 2

Pixel ke-	Proses 1	Kunci	Hasil
1	AC|JH|KO	RA|MA|DA	RC|VH|NO
2	BB|MN|OP	NR|AM|AD	OS|MZ|OS
3	CD|PO|GA	AN|RA|MA	CQ|GO|SA
4	JK|PL|OL	DA|NR|AM|	MK|CC|OX
5	AN|NA|PC	AD|AN|RA	AQ|NN|GC
6	IK|LF|GD	MA|DA|NR	UK|OF|TU
…	…	…	…
n	OL|PB|DE	MA|DA|NR	AL|SB|QV

2.2 Implementasi Algoritma Enkripsi

Proses enkripsi merupakan proses penyandian pesannya. Karana pesan citra bisa dibaca dalm bentuk *array of byte*, maka proses enkripsinya pun dilakukan dari *byte* ke *byte*. Kerangka algoritma program untuk proses enkripsi sebagai berikut:

```
function encryption (input : message: int[], key: int[]) → int[]
{dalam implementasi nyatanya, message dan key merupakan suatu array of byte, message: merupakan pesan yang akan dienkripsi, key: kunci yang digunakan untuk enkripsi}

int[] returned
int temp1
int temp2
{returned, temp1 dan temp2 dalam implementasi programnya adalah byte, returned adalah variabel untuk menyimpan total hasil enkripsi, temp1 dan temp2 adalah hasil enkripsi untuk tiap-tiap iterasi}

int keyIn
int i ← 0
int Lengthkey ← panjang key
int Lengthmessage ← panjang message

while (i< Lengthmessage)
{belum end of file, Message dalam bentuk byte}
    keyin ← key[2*i mod Lengthkey]-65
    {nilai key dikurangi dengan angka 65 karena dalam ASCII karakter alfabet dimulai dari karakter ke 65}

    temp1 ← message[i] / 16
    {dalam implementasi programnya, message[i] harus dikonversi dulu dari byte menjadi integer}

    temp1 ← temp1 + keyin
    temp1 ← (temp1 mod 26) + 65
    {hasilnya ditambah lagi dengan nilai 65 untuk mendapatkan karakter ASCII yang sesbenarnya}
    keyin←key[2*i+1 mod Lengthkey]-65
    temp2 ← message[i] mod 16
    temp2 ← temp2 + keyin
    temp2 ← (temp2 mod 36) + 65
    returned[2*i] ← temp1
    returned[2*i+1] ← temp2

→ returned
```

Dilihat dari algoritma enkripsinya bisa ditentukan bahwa ukuran teks hasil enkripsi akan menjadi lebih besar 2 kali lipat daripada ukuran citranya. hal ini dikarenakan tiap 1 *byte* pesan dienkripsi menjadi 2 alfabet (2 *byte*) pesan.

2.3 Implementasi Algoritma Dekripsi

Proses dekripsi merupakan proses rekonversi dari teks hasil enkripsi menjadi pesan yang sebenarnya kemudian dikonversi lagi menjadi citra yang dimaksud. Kerangka algoritma program untuk proses dekripsi sebagai berikut:

```
function decryption (input : message: int[], key: int[]) → int[]
{dalam implementasi nyatanya, message dan key merupakan suatu array of byte, message : merupakan pesan yang akan dienkripsi, key: kunci yang digunakan untuk enkripsi}

int[] returned
int temp
{returned, temp dalam implementasi programnya adalah byte, returned adalah variabel untuk menyimpan total hasil dekripsi, temp adalah hasil dekripsi untuk tiap-tiap iterasi}

int in1
int in2
int keyIn
int i ← 0
int Lengthkey ← panjang key
int Lengthmessage ← panjang message

while (i< Lengthmessage/2)
{belum end of file, Message dalam bentuk byte}
    keyIn ← key[2*i mod Lengthkey]-65
    {nilai key dikurangi dengan angka 65 karena dalam ASCII karakter alfabet dimulai dari karakter ke 65}

    in1 ← message[i*2]-65
    {dalam implementasi programnya, message[i] harus dikonversi dulu dari byte menjadi integer, nilainya dikurangi dengan 65 untuk mendapatkan nilai aslinya}

    in1 ← in1 - keyIn
    in1 ← in1 mod 26

    keyIn ← key[2*i+1 mod Lengthkey]-65
    in2 ← message[2*i+1]-65
    in2 ← in2-keyIn
    in2 ← in2 mod 26
```

```
    temp ← (in1*16) + in2
    returned[i] ← temp

→ returned
```

3. HASIL DAN PEMBAHASAN

3.1 Enkripsi Berbagai Jenis Gambar

Setelah dilakukan simulasi dengan menggunakan aplikasi program, didapatkan beragam hasil. Secara umum hasil tersebut bisa diklasifikasikan menjadi 2 jenis untuk dianalisis, yaitu untuk citra dengan distribusi warna heterogen dan citra dengan distribusi warna cenderung homogen.

3.1.1 Enkripsi gambar dengan warna heterogen

Proses enkripsi untuk citra dengan distribusi warna heterogen akan menghasilkan *chipertext* dengan distribusi huruf yang heterogen juga. Sebagai contoh enkripsi untuk citra pada gambar 1 di bawah ini:

Gambar 1: contoh citra dengan distribusi warna heterogen, tipe jpg, ukuran 77,5 kb

menghasilkan *chipertext* (potongan) sebagai berikut :

```
.....WDGTOHNVBUSDKVIGWLQHAB
VYFUTNQMZURURUQMRJRYMZEOWSANBDGTGURVC
QUTQUQTWSOQPWRUFPVHWSUAQVESRLVSABJBTU
QNNNAXNULGWVNGTBQGQRWJSSRRUUNSLGVPZQU
USPHISIBSVDTVHGYSZZSBQGKOBLZXHYQTPTAK
ZYJIHOLPASRCSZMVRLCVMOGDQVWQBINOGTLHI
GUSNUOZELTLCKUYTFSTUMSLMBNUSULNCOPVVI
EGNPYVBSTYLRWGTVSBOYRKKLXVWAUEPIOMYON
CUUNRSTVVSDVZNOPNZIBXVFLOPNYLVUQWETWH
ZUXCLWTRSICUBVUIGVVMYPQRIIJMKHBQBBIBR
VTTCTSSGCKQRJYTWPVHNRPHTLVDRBTGMQBVBR
LGTKSVYIVDUXPNVHQOYCIVJGQGWKQDLBNUJGB
NQYIZGMRKNOPBOHMQLPQQSCUCQOOVCVZQTDQO
OLXLUTTELNQMBQZYPERSWUTUADVXNVWLNQUVQ
TQSSSNMBYQWLTPTQMQPIXMTMSRUAXHITPJLYG
TORVIPWQPKSZSAMQPHBPWGIOHVHPQGZFKQHXH
XPKVGJJKOLRQSUKIKJQKTQOVRHLJNNOTPVJJW
GBHBPMWLJKLOMWQRVMJJMRKVRLXKKLLZIWRNX
HMINAIRTUVKLWIYJUSSWRJOLBJBRKZGNQKXKU
TRXNLTKAKASVXOLSLZLZTUYTLMMTMTUOZNMRM
YMYUTZSMWMQNQVLAKNONVNVVQAPNTNANAVVAF
HHGNIMQGUEIHKNKMRGXELHMNNMVGZELKLNHMO
GVFHHJPHPPHVEKLHSHSPNUGIIJNLPQOUNHIKX
HXPKVGJTHSIZPVTHJJKTIAPNVMIQIRJRRHYKJ
OJQKBPIYEMIMXIXQOWENJMQLYQTVIKQJSKWRH
AKKLLSLPUNXKLVICIXROXQJUJULARVWIMQKXK
UTRXNLTKAKASVXOLSLZLZTUYTLMMTMTUOZNMR
MYMYUTZSMWMPNPVKAJNNNUNUVPAONSNZNBDTD
EGHSNJMPGTEIIHNJNPGTHVHGBPTTKIOQNKWMW
ZPFOLJMQHYDJEJQRJXGAVKIFIPRVPTAJGMTWK
ATPUHAIPKGPGPSMHRTWSYTZDPCPLLNWJOAIUI
GQGNSXCLURVMUOVBZQYTGPHWHBOHEPMMPVJXA
GTLKIHZOAULYOQKGSOYUKUSNTKXPRTNTMTKPR
OASHAIOISVLMAQYHJRJTIPWUTHLNTSJRRIEFQ
NQUSRSVAEGMLOVRWRASPKONVUXOYEJQVRPOSN
DMJLUCTVSNRSVJTOYUJFOSKOVVYBIFOMMSRNR
ATHNHIUPHAZVGJMIOUVAROFLQNJBRPBIWFNUN
TVZYHGPJMRASMPTFLNKRSVTCTALVJLSPSVTCE
RPJNLTOUVEVUHAQZRGWSPOTOJBXQBFOKTQMZY
JDIUJTNVXTPBTMKMXQYZRGISNSMA.........
```

3.1.2 Enkripsi gambar dengan warna homogen

Proses enkripsi untuk citra dengan distribusi warna homogen akan menghasilkan *chipertext* dengan distribusi huruf yang cenderung homogeny juga (banyak pengulangan string). Sebagai contoh enkripsi untuk citra pada gambar 1 di bawah ini:

Gambar 2: contoh citra dengan distribusi warna cenderung homogen, tipe .gif, ukuran 14,5 kb

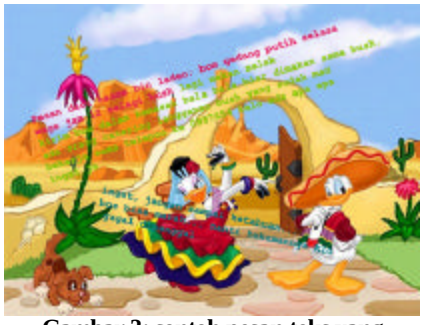
menghasilkan *chipertext* (potongan) sebagai berikut:

```
.....TMOIDEGHGNGMWGTENHHOIMOGWEGHGN
GMPGTEHH
ODOYEJLHXNUZQZPOQNSQUDGYRGULWPUSHTSKU
UWNRWUATIHJVJNUVVSPVGNGMPHDEGHLNGMOGT
EHHGNGMOMVEHIRNGMTGTEGHGNHMOGTEGNQNHO
WGTEJHGNGMOGUEGHINGMOGUHHHGNIMOGTEGIR
NGMOGTLIHHQIMOGVEGHGNGNSGTEGHGVTUVMCE
GHKNGMOGTEHHGNGMOQXEGHGNGZOGTEGHGROMO
GTEGHHNGMOGTIOHGNGMOGUIHNKTVSQMYGGMGT
OSDNXKVOJTOSDNTGGLJSJOOLAKPNUTKSDNALJ
HGQIPOJTHNKQQHPOJDHIKLPGPOJBHQKIQJPYJ
YHGHGNGMOGTEGHJXGMPGTEJHGNGMOGUEGHHNG
MOQTEIHGNKMOGTEGHHNGMOGUNGRGNJMOGXEGH
GNGMPGTEGHHPSMOGTEGHGNGMOGTEMHHNJMOGW
EGHGNGMPGTEMHGNGMPHDEGHLNGMOGTEHHGNGM
PHHEHIRNGMTGTEGHGNHMOGTEHJMNHOWGTEJHG
NGMOGUEGHINGMOGVEHHGNKMOGTEGHHNGMOODO
YEJLHXNUZQZPOQNSQUDGYRGULWPUSHTSKUUWN
RWUATIHJVJNUVVSPVGUGUHINIMOGXEGHGNGMP
GTEGJKOUMOGTEGHGNGMOGTIOHGNGMOGUEGHGN
GQWGTEGHGNHBDTBTVVGNGNOKDIMLPRMMOGUEI
HHNGQWGTIOHGNGBDUGEGHSRHSSMIKINLSVQRK
GEGHHCVACGTEULHTKSDMVKLHGTKUOGTEGHGNH
BDTEEGPKNGMAGBEOHONPMWGFEPHPNSNPGEEQH
ROHNTGIESHSNVNTHBFJIJOLNRHWFOIHNSMAGF
ESHSNSNPGFESHSNSMAGFESHSNSMAGFESHSNSM
AGFESHSNSMAGFESHSNSMAGFESHSNHMBGEERHT
NUMBHTEUHUOGNSGHEUHODOYEJLHXNUZQZPOQN
SQUDGYRGULWPUSHTSKUUWNRWUATIHJVJNUVVS
PVUOKNSGHEUHUNUNSHUESHSNSMAGFFHIHNSMA
GFESHSNSNPGFESHSNSMAGFESHSNSMIBTVKIPU
```

TBGDKLRJYHZUTTODOYEJLHXNUZQZPOQNSQUDG
YRGULWPUSHTSKUUWNRWUATIHJVJNUVVSPVOJP
JZOWSKWQQQKNSZRQAKSTUTL……

3.1.3 Enkripsi teks yang dimanipulasi sebagai gambar

Pada bab pendahuluan disebutkan bahwa teknik ini juga dapat digunakan pada pesan citra yang seolah-olah pesan teks dengan membuat sebuah citra yang isinya adalah teks menggunakan editor gambar. Dengan menggunakan prinsip heterogenitas yang disebutkan di atas, pesan ini juga harus dibuat sedemikian rupa sehingga terlihat heterogen. Proses enkripsi untuk teks yang sudah dimanipulasi sebagai bentuk citra tidak akan jauh berbeda hasilnya dengan proses enkripsi untuk jenis gambar sebelumnya. Sebagai contoh enkripsi untuk citra di bawah ini:



Gambar 3: contoh pesan teks yang dimanipulasi sebagai citra, tipe .jpeg, ukuran 77,5 kb

Menghasilkan *chipertext* (potongan) sebagai berikut:

……
WDGTOHNVBUSDKVIGWLQHABVYFUTNQMZURURUQ
MRJRYMZEOWSANBDGTGURVCQUTQUQTWSOQPWRU
FPVHWSUAQVESRLVSABJBTUQNNNAXNULGWVNGT
BQGQRWJSSRRUUNSLGVPZQUUSPHISIBSVDTVHG
YSZZSBQGKOBLZXHYQTPTAKZYJIHOLPASRCSZM
VRLCVMOGDQVWQBINOGTLHIGUSNUOZELTLCKUY
TFSTUMSLMBNUSULNCOPVVIEGNPYVBSTYLRWGT
VSBOYRKKLXVWAUEPIOMYONCUUNRSTVVSDVZNO
PNZIBXVFLOPNYLVUQWETWHZUXCLWTRSICUBVU
IGVVMYPQRIIJMKHBQBBIBRVTTSTUMSLMBNUSU
LNCOPVVIEGNHTLVDRBTGMQBVBRLGTKSVYIVDU
XPNVHQOYCIVJGQGWKQDLBNUJGBNQYIZGMRKNO
PBOHMQLPQQSCUCQOOVCVZQTDQOOLXLUTTELNQ
MBQZYPERSWUTUADVXNVWLNQUVQTQSSSNMBYQW
LTPTQMQPIXMTMSRUAXHITPJLYGTORVIPWQPKS
ZSSTUMSLMBNUSULNCOPVVIEGNZFKQHXHXPKVG
JJKOLRQSUKIKJQKTQOVRHLJNNOTPVJJWGBHBP
MWLJKLOMWQRVMJJMRKVRLXKKLLZIWRNXHMINA
IRTUVKLWIYJUSSWRJOLBJBRKZGNQKXKUTRXNL
TKAKASVXOLSLZLTUYTLMMTMTUOZNMRMYMYUT
ZSMWMQNQVLSTUMSLMBNUSULNCOPVVIEGNVAFH
HGNIMQGUEIHKNKMRGXELHMNNMVGZELKLNHMOG
VFHHJPHPPHVEKLHSHSPNUGIIJNLPQOUNHIKXH
XPKVGJTHSIZPVTHJJKTIAPNVMIQIRJRRHYKJO
JQKBPIYEMIMXIXQOWENJMQLYQTVIKQJSKWRHA
KKLLSLPUNXKLVICIXROXQJUJULARVWIMQKXKU
TRXNLTKAKASVXOLSLZLTUYTLMMTMTUOZNMRM
YMYUTZSMWMPNPVKAJNNNUNUVPAONSNZNBDTDE
GHSNJMPGTEIIHNJNPGTHVHGBPTTKIOQNKWMWZ
PFOLJMQHYDJEJQRJXGAVKIFIPRVPTAJGMTWKA
TPUHAIPKGPGPSMHRTWSYTZDPCPLLNWJOAIUIG
QGNSXCLURVMUOVBZQYTGPHWHBOHEPMMPVJXAG
TLKIHZOAULYOQKGSOFRYUKUSNTKXPRTNTMTKP
ROASHAIOISVLMAQYHJRJTIPWUTHLNTSJRRIEF
QNQUSRSVAEGMLOVRWRASPKONVUXOYEJQVRPOS
KOVVYBIFOMMSRNRATHNHIUPHAZVGJMIOUVARO
FBIWFNUNTVZYHGPJMRASMPTFLNKRSVTCTALVJ
LSPSVTCERPJNLTOUVEVUHAQZRGWSPOTOJBXQB
JDIUJTNVXTPBTMKMXQYZRGISNSMA.....

Dari hasil-hasil yang didapatkan di atas terlihat bahwa citra dengan tingkat homogenitas warna sangat tinggi akan menghasilkan *chipertext* dengan tingkat homogenitas penyebaran hurufnya juga tinggi. Artinya banyak terjadi pengulangan string yang mengakibatkan *chipertext* mudah dianalisis dan dipecahkan kuncinya. Oleh karena itu jika ingin menggunakan teknik ini untuk pesan citra rahasia, gunakanlah pesan citra dengan distribusi warna sangat heterogen. Cara menentukan apakah citra termasuk heterogen atau homogen sangat mudah dengan melihat dari warna-warni citra tersebut secara kasat.

Hal ini bisa menjadi kekurangan sekaligus kelebihan. Ukuran yang semakin besar menyebabkan proses pengiriman pesan akan menjadi lebih lama. Tapi di sisi lain hal ini akan lebih menyulitkan kriptanalis untuk menganalisis. Selain itu dengan besarnya ukuran *chipertext*, jika terjadi kerusakan nonteknis (umumnya kerusakan akibat hal nonteknis terjadi hanya beberapa *byte*), citra masih bisa dipahami secara keseluruhan (kerusakannya tidak terlalu signifikan untuk menjadikan penerima pesan melakukan kesalahan dalam mempersepsikan citra).

3.2 Serangan Kriptografi

Algoritma ini memiliki kelebihan yang menyulitkan orang-orang yang berusaha memecahkan kodenya. Telah dijelaskan di atas bahwa teknik analisis citra sulit dilakukan karena ukurannya yang besar dan heterogenitasnya. Penyesatan yang mendorong kriptanalis menganggap bahwa *chipertext*nya adalah sebuah pesan teks akan menyulitkan lagi. Namun kalaupun k riptanalis mengetahui bahwa *chipertext*nya adalah hasil enkripsi dengan teknik ini, kemudian kriptanalis mencoba memecahkan dengan menggunakan kunci secara *brute force*, teknik ini masih memiliki kekuatan, yaitu jika sembarangan kunci dicoba, akan ada kunci yang menyebabkan kinerja komputer melambat atau hasil dekripsi tidak bisa dibaca komputer. Penjelasannya sebagai berikut, misalkan hasil dekripsi *chipertext* dengan sembarang kuncinya ditunjukkan dengan tabel di bawah ini:

Tabel 3: contoh dekripsi yang menyebabkan error

chipertext	*key*	Dekripsi	Konversi ke desimal
VC	CA	SB	289
OS	NT	AA	0
KQ	IK	BF	21

untuk *chipertext* VC yang didekripsi dengan kunci CA akan menghasilkan bilangan desimal 288. Angka ini tidak bisa dikonversi ke *byte* untuk menghasilkan citra karena batas konversi ke *byte*-nya hanya dari 0-255.

4. KESIMPULAN

Teknik konversi citra ke dalam teks menambah jumlah teknik kriptografi pesan. Teknik ini bisa dipakai karena memiliki beberapa kelebihan, yaitu:

- Algoritmanya mudah tetapi pemecahannya sulit.
- Proses dekripsi dengan menggunakan kunci tertentu yang salah bisa memperlambat proses pemecahan.
- Jika pesan rusak, tidak akan menyebabkan kerusakan yang fatal.

Selain itu ada beberapa kekurangan yang terjadi, yaitu:

- Ukuran chipertext menjadi lebih besar 2 kali lipat.
- Penggunaan hanya bagus untuk citra dengan tingkat heterogenitas warna tinggi

DAFTAR REFERENSI

[1] Munir, Rinaldi, "*Diktat Kuliah IF15054, Kriptografi*", 2006, Bandung