IMPLEMENTASI METODE SUPER ENKRIPSI (*VINEGERE CIPHER –ARNOLD CAT MAP*) PADA MATRIKS CITRA

SKRIPSI

OLEH
CICI ERISA MAULIDAH
NIM. 14610059

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018

IMPLEMENTASI METODE SUPER ENKRIPSI *(VINEGERE CIPHER- ARNOLD CAT MAP)* PADA MATRIKS CITRA

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Cici Erisa Maulidah
NIM. 14610059

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018

IMPLEMENTASI METODE SUPER ENKRIPSI
(*VINEGERE CIPHER –ARNOLD CAT MAP*) PADA MATRIKS CITRA

SKRIPSI

Oleh
Cici Erisa Maulidah
NIM. 14610059

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 08 November 2018

Pembimbing I,	Pembimbing II,
Dr. H Turmudi, M.Si, Ph.D	Muhammad Khudzaifah, M.Si
NIP. 19571005 198203 1 006	NIP. 19900511 20160801 1 057

Mengetahui,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

IMPLEMENTASI METODE SUPER ENKRIPSI
***(VINEGERE CIPHER-ARNOLD CAT MAP)* PADA MATRIKS CITRA**

SKRIPSI

Oleh
Cici Erisa Maulidah
NIM. 14610059

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 28 November 2018

Penguji Utama	:	H. Wahyu H. Irawan, M. Pd	.....
Ketua Penguji	:	Mohammad Jamhuri, M. Si	.....
Sekretaris Penguji	:	Dr. H. Turmudi, M.Si, Ph. D	.....
Anggota Penguji	:	Muhammad Khudzaifah, M.Si	.....

Mengetahui,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Cici Erisa Maulidah

NIM : 14610059

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Metode Super Enkripsi *(Vinegere Cipher- Arnold Cat Map)* Pada Matriks Citra

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 28 November 2018
Yang membuat pernyataan,

Cici Erisa Maulidah
NIM. 14610059

MOTTO

“Jika kamu menginginkan sesuatu, kamu akan menemukan caranya. Namun jika serius, kau hanya menemukan alasan”

-Jim rohn-

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Ayahanda Sugeng Hadi Purwanto dan Ibunda Khotima Tusifak tercinta, yang senantiasa dengan ikhlas mendoakan, memberi nasihat, semangat, dan kasih sayang yang tak ternilai, serta Adik tersayang Kiki Nur Alvin yang selalu megnhibur dan Keluarga yang selalu menjadi inspirasi bagi penulis.

.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Swt atas rahmat, taufik serta hidayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, penulis banyak mendapat bimbingan dan arahan dari berbagai pihak. Untuk itu ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis sampaikan terutama kepada:

1. Prof. Dr. H. Abdul Haris, M.Ag, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Usman Pagalay, M.Si, selaku ketua Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. H. Turmudi, M. Si, Ph.D, selaku dosen pembimbing I yang telah memberikan arahan dan berbagi ilmunya kepada penulis.
5. Muhammad Khudzaifah, M.Si, selaku dosen pembimbing II yang telah banyak memberikan arahan, nasihat, motivasi dan berbagi pengalaman kepada penulis.
6. Segenap sivitas akademika Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terima kasih atas segala ilmu dan bimbingannya.

7. Bapak dan Ibu serta kakak tercinta yang selalu memberikan do'a, semangat, serta motivasi kepada penulis sampai saat ini.
8. Sahabat-sahabat terbaik penulis yang selalu menemani, membantu, dan memberikan dukungan sehingga penulis dapat menyelesaikan skripsi ini.
9. Seluruh teman-teman di Jurusan Matematika angkatan 2014 (MATH EIGEN) khususnya Matematika-B, Teman-teman Kos, Kelompok KKM dan ABA 44 yang berjuang bersama-sama untuk meraih mimpi, terimakasih kenang-kenangan indah yang dirajut bersama dalam menggapai impian.
10. Semua pihak yang tidak dapat disebutkan satu-persatu, yang telah membantu penulis dalam menyelesaikan skripsi ini baik moril maupun materiil.

Semoga Allah Swt melimpahkan rahmat dan karunia-Nya kepada kita semua. Akhirnya penulis berharap semoga dengan rahmat dan izin-Nya mudah-mudahan skripsi ini bermanfaat bagi penulis dan bagi pembaca. *Amiin.*

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 28 November 2018

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Maulidah, Cici Erisa. 2018. **Implementasi Metode Super Enkripsi (*Vinegere Cipher – Arnold Cat Map*) Pada Matriks Citra.** Skripsi. Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I): Dr. H. Turmudi M.Si, Ph. D, Pembimbing (II): M. Khudzaifah M.Si.

Kata Kunci: Matriks, Enkripsi, Dekripsi, Super Enkripsi, Citra, ACM, *Vinegere*.

Enkripsi merupakan proses menyandikan pesan menjadi tak terbaca(acak) dan dekripsi merupakan proses kebalikannya. pesan yang dienkripsi berbentuk teks, namun penelitian ini akan menggunakan pesan gambar sebagai objek enkripsi. Penelitian ini bertujuan mengetahui proses enkripsi dan dekripsi pada matriks citra menggunakan metode Super Enkripsi.

Super enkripsi merupakan algoritma enkripsi yang menggabungkan dua metode yaitu subsitusi dan transposisi. Subsitusi bertujuan untuk merubah setiap entri melalui operasi matematika dengan kunci yang telah ditentukan, entri dalam enkripsi teks berupa karakter dan dalam enkripsi citra berupa nilai pixel. Selanjutnya, transposisi digunakan dalam merubah/ menggeser posisi tiap entri. Pada penelitian ini, algoritma *Vinegere cipher* sebagai implementasi metode subsitusi dan algoritma *Arnold Cat Map* akan berperan sebagai enkripsi berbasis transposisi.

Algoritma enkripsi metode super enkripsi terbentuk dengan menjalankan *Vinegere cipher* terlebih dahulu, diikuti dengan *Arnold Cat Map*. *Vinegere cipher* dengan operasi *Exclusive-OR* merupakan operasi *bit-string* dalam lingkup digital, operasi *bit-string* tersebut akan mengubah setiap nilai pixel yang ada pada *plain-image*(gambar asli). Selanjutnya *Arnold Cat Map* melakukan transformasi pixel, pixel *plain-image* yang memiliki koordinat (x, y) akan di petakan pada koordinat baru (x', y') dengan melakuakan perkalian matriks koordinat dengan matriks ordo 2×2 yang merupakan bentukan dari algoritma *Arnold Cat Map*.

Pada algoritma dekripsi akan dimulai dengan menjalankan *Arnold Cat Map* dan diikuti *Vinegere cipher*, proses ini merupakan kebalikan dari enkripsi. Dekripsi *Arnold cat map* akan melakukan transformasi titik pixel koordinat(x', y') dari *cipher-image* (gambar hasil enkripsi) sehingga terbentuk koordinat (x, y) yang merupakan koordinat asal sebelum dienkripsi. Selanjutnya dekripsi *Vinegere cipher* akan mengubah tiap nilai pixel dalam *cipher-image* dengan operasi *bit-string* sehingga nilai pixel akan kembali sama seperti nilai pixel pada *plain-image*.

Penggunaan super enkripsi dengan *Vinegere cipher* dan *Arnold Cat Map* akan mendapatkan keamanan ganda. Keamanan pertama terletak pada setiap nilai pixel yang berubah sesuai kunci berupa *password* yang dimasukkan. *Password* yang digunakan terdiri minimal tiga karakter, namun semakin banyak karakter yang digunakan maka gambar akan semakin terenkripsi dengan baik. Selanjutnya keamanan kedua, pixel gambar akan tersebar sesuai dengan transformasi yang dilakukan oleh *Arnold Cat Map (ACM)*.

ABSTRACT

Maulidah, Cici Erisa. 2018. **The Implementation of Super Encryption Method (Vinegere Ciphers-Arnold Cat Map) in Image Matrix.** thesis. Mathematic, Faculty of Sains and Technology, Maulana Malik Ibrahim State Islamic University Of Malang. Supervisor (I) : Dr. H. Turmudi M.Si, Ph. D , Supervisor (II): M. Khudzaifah M.Si.

Kata Kunci: Matrix, Encryption, decryption, Super Encryption, Image, ACM, Vinegere cipher.

Encryption involves the code for messages as illegible reverse (random) and decryption is involves the contrary. Generally, the encrypted message shaped is a text, but this research is going to use images as objects encryption. This research aims to know the encryption and decryption process on image matrix uses super encryption method.

Super encryption is encryption algorithm that combines two methods namely substitution and transposition. The aim of substitution to change any entry through a mathematical operation using determined key. The input of the text encryption is characters and the input of an image encryption is pixel. Next, the transposition is used in change/shifting the position of every entry. In this research, the vinegere ciphers algorithm as the implementation of substitution methods and the algorithms Arnold Cat Map will serve as based transposition.

Algorithms encryption of super encryption method is formed by running vinegere ciphers as first step, and then followed by arnold cat map. Vinegere cipher with Exclusive-OR operation is bit-string exclusive-or operating within the scope of digital, bit-string operation will change any value of pixel in plain-image (the original image). Accordingly Arnold Cat Map transforms the pixel, the plain-image pixel having coordinates (x, y) will be mapped into new coordinates (x', y') by multiplying coordinate matrices with matrix order 2×2 which are formed Arnold Cat Map algorithm.

Decryption algorithm initiated by running Arnold Cat Map and followed Vinegere ciphers, this process is the inverse of encryption. The Description of Arnold Cat Map will transform the pixel of coordinate (x', y') from cipher-image (result from encryption image). Such a way (x, y) that is original coordinate before encryption is formed emerged. Next, Vinegere cipher description will change each pixel value in cipher-image with bit-string operation so pixel value will return to pixel value in plain-image.

The usage of super encryption with Vinegere cipher and Arnold Cat Map will produce double security. The first security is located in each pixel value that change to suitable key is the form of a password. The password that used consists at least three characters. But, if more characters used then image will be more well encrypted. For the second security, pixel of image will spread that suitable with transformation by Arnold Cat Map.

الملخص

اريسه ماوليداه, سيسي. ٨ ٢٠١. **تنفيذ طريقة التشفير الفائق (Arnold – Vinegere cipher cat map) في مصفوفة الصورة** .بحث الجامعي. شعبة الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك ابراهيم الإسلامية الحكومية مالانج . المستشارين: (١) الدكتور تورمودي الماجستير (٢) محمد خودزايفاه الماجستير

الكلمات الرئيسية : المصفوفة، التشفير، فك التشفير، الفائق مشفر، صورة، ACM، Vigenere cipher

يتضمّن التشفير رمز الرسائل كعكس غير مقروء (عشوائي) ، وينطوي فك التشفير على عكس ذلك. عموما رسالة مشفرة على شكل نص ، ولكن هذا البحث سوف يستخدم الصور كأداة تشفير الكائنات. يهدف هذا البحث إلى معرفة عملية التشفير وفك التشفير على مصفوفة الصور التي تستخدم طريقة التشفير الفائق.

التشفير الفائق هو خوارزمية التشفير التي تجمع بين طريقتين هما الاستبدال والتبديل. يهدف الاستبدال إلى تغيير أي إدخال من خلال عملية حسابية بواسطة مفاتيح يتم تعيينها ، وإدخال في نموذج تشفير النص الخاص بالأحرف ودخول شكل تشفير صورة البكسل. بعد ذلك ، تبديل يستخدم في تغيير / تحويل موقف كل إدخال.في هذا البحث ، فإن خوارزمية vinegere cipher مثل تنفيذ أساليب الاستبدال والخوارزميات arnold map cat ستخدم على أساس التحويل.

خوارزميات تشفير طريقة تشفير فائقة تتكون عن طريق تشغيل vinegere cipher كخطوة أولى ، ثم تليها arnold cat map. تشفير الكرمة مع عملية Exclusive-OR هي سلسلة ذات حصرية – أو تعمل ضمن نطاق رقمي ، ستعمل عملية سلسلة البت على تغيير أي قيمة متوفرة في الصورة العادية (الصورة الأصلية). الخطوة التالية خريطة أرنولد القط لتحويل البيكسل ، سيتم تعيين صورة بياض عابرة ذات إحداثيات (x,y) على إحداثيات جديدة (x',y') عن طريق مضاعفة تنسيق المصفوفات مع المصفوفة بالترتيب 2×2 أشكال خوارزمية Arnold cat map.

على خوارزمية فك التشفير ستبدأ بتشغيل Arnold cat map وتتبع Vinegere ciphers ، هذه العملية هي عكس التشفير. سيؤدي وصف Arnold cat map إلى تحويل بكسل الإحداثيات (x',y') من image- cipher (ناتجة عن صورة التشفير). لذلك تنسيق إحداثيات الأصلي قبل أن

يتم تشكيل التشفير (x, y). بعد ذلك ، سيؤدي التشفير Vinegere cipher المتغير إلى تغيير قيمة كل بكسل في صورة cipher-image مع تشغيل سلسلة البت بحيث ستعود قيمة البكسل إلى قيمة البكسل في صورة عادية.

سيحصل استخدام التشفير الفائق مع Vinegere cipher وArnold Cat map على حماية مزدوجة. يقع الأمان الأول في كل قيمة بكسل تتغير إلى مفتاح مناسب. عل شكل كلمة مرور. تتكون كلمة المرور المستخدمة من ثلاثة أحرف على الأقل. ولكن ، إذا تم استخدام المزيد من الأحرف ، فستكون الصورة أكثر تشفيرًا بشكل جيد. بالنسبة للأمن الثاني ، سيتم نشر بكسل الصورة المناسبة مع التحويل الذه تم فيامحاب Arnold Cat map.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era sekarang ini bentuk komunikasi mengalami banyak perkembangan. Hal ini terlihat jelas dari cara masyarakat memanfaatkan berbagai perangkat digital sebagai alat komunikasi. Melalui alat komunikasi digital masyarakat dapat berkomunikasi jarak jauh baik melalui suara, pesan teks, video dan gambar (citra). Banyak berbagai jenis gambar tersebar dilingkup digital dan bisa jadi salah satu gambar adalah pesan gambar yang dikirim hanya untuk pihak tertentu yang bersifat rahasia. Hal ini bisa merugikan pihak pemilik gambar baik sebagai privasi maupun finansial.

Menjaga amanah dan melindungi aib bisa menjadi salah satu tujuan dari merahasiakan data. Dalam Al-Qur'an surat Al-Hujarat/49:12, telah dijelaskan pentingnya melindungi rahasia.

يأَيُّهَا ٱلَّذِينَ ءَامَنُواْ ٱجتَنِبُواْ كَثِيرا مِّنَ ٱلظَّنِّ إِنَّ بَعضَ ٱلظَّنِّ إِثم وَ لَا تَجَسَّسُواْ وَلَا يَغتَب بَّعضُكُم بَعضًا أَيُحِبُّ أَحَدُكُم أَن يَأكُلَ لَحمَ أَخِيهِ مَيتا فَكَرِهتُمُوهُ وَٱتَّقُواْ ٱللَّهَ إِنَّ ٱللَّهَ تَوَّاب رَّحِيم

"Wahai orang-orang yang beriman! jauhilah kebanyakan dari prasangka, karena sesungguhnya sebagian dari prasangka itu adalah dosa dan janganlah mengintip atau mencari-cari kesalahan orang lain dan janganlah dari kamu mengumpat sebagian yang lain. Apakah seseorang dari kamu suka memakan daging saudaranya yang telah mati? Maka sudah tentu kamu jijik kepadanya" (QS. Al-Hujarat '/49:12).

Dalam surat di atas tersirat "janganlah mengintip" kalimat ini mengartikan tentang penggunaan indera mata. Jangankan menceritakan bahkan mengintip ketidakbaikan orang lain (aib) sangat dilarang dalam agama Islam. Maka untuk menghindari keburukan yang tersebar dalam jaringan internet,

diperlukan solusi agar rahasia tetap terjaga dan dapat sampai kepada pihak yang dituju.

Salah satu teknik keamanan pesan yang banyak dipakai sampai saat ini adalah kriptografi. Enkripsi dan dekripsi merupakan dua fungsi yang ada dalam kriptografi. Enkripsi citra bertujuan menyandikan citra (*plain-image*) sehingga tidak dapat dikenali lagi (*cipher-image*) (Munir, 2012). Metode enkripsi citra yang sekarang ini banyak dikembangkan adalah fungsi *Chaos*. *Chaos* merupakan fungsi yang bersifat acak dan peka terhadap nilai awal. Suryadi (2014), menggunakan skema transposisi berbasis fungsi *Chaos* dalam melakukan enkripsi citra digital dengan algoritma *Arnold Cat Map*. Kunci yang digunakan adalah dengan menggunakan iterasi dan dua variabel lain. Namun algoritma *Arnold Cat Map* memiliki kelemahan dalam hal iterasi. Enkripsi citra yang hanya mengunakan *Arnold Cat Map* (ACM) dianggap tidak aman karena sifat periodiknya yang dapat mengembalikan citra asli melalui serangan *brute force* artinya nilai parameternya dapat ditemukan dengan mudah (Purba, 2014).

Perkembangan penelitian selanjutnya dilakukan oleh Rinaldi munir dengan menggabungkan dua teknik chaos yaitu ACM dan *Logistic Map*. *Logistic Map* merupakan algoritma enkripsi bersifat *chaostic*, dengan menggunakan nilai acak. Nilai-nilai acak yang dihasilkan dari persamaan logistic tidak pernah berulang kembali sehingga *Logistic Map* dikatakan tidak mempunyai periode (Munir, 2012).

Penggabungan dua algoritma enkripsi juga di lakukan oleh Gondo Suwiryo dan rekannya dari Universitas Brawijaya Malang. Penelitiannya dipublikasikan dalam jurnalnya yang berjudul Enkripsi Citra Digital

Menggunakan *Vinegere Cipher* Dan *Logistic Map*. *Vinegere cipher* adalah teknik enkripsi yang merupakan salah satu algoritma kriptografi yang digunakan untuk penyandian teks. Namun dalam penelitiannya pemakaian *vinegere cipher* diperluas dari teks ke citra bitmap 24-bit.

Berdasarkan pada penelitian-penelitian yang telah dilakukan sebelumnya, penelitian ini akan mengembangkan dua teknik enkripsi citra berbasis transposisi dan substitusi. Algoritma *Arnold Cat Map* akan berperan sebagai enkripsi berbasis transposisi, setiap pixel dikoordinat (x, y) akan di petakan pada koordinat baru (x', y'). Selanjutnya *Vinegere cipher* yang memiliki algoritma berbasis substitusi, akan difungsikan untuk merubah nilai setiap pixel. Penggabungan teknik transposisi dan substitusi ini disebut juga dengan super enkripsi. Dengan super enkripsi algoritma enkripsi yang diciptakan lebih kompleks karena selain nilai pixel gambar berubah, pixel citra juga tersebar secara acak.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang tersebut maka rumusan masalah dalam penelitian ini adalah

1. Bagaimana proses enkripsi pada matriks citra menggunakan metode Super Enkripsi?
2. Bagaimana proses dekripsi pada matriks citra menggunakan metode Super Enkripsi?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas maka tujuan penelitian adalah

1. Mengetahui proses enkripsi pada matriks citra menggunakan metode Super Enkripsi.
2. Mengetahui proses dekripsi pada matriks citra menggunakan metode Super Enkripsi.

1.4 Manfaat Penelitian

Hasil penulisan ini diharapkan mampu memberikan manfaat pada pembaca umumnya dan penulis khususnya, selain itu di harapkan:

1. Dengan bantuan komputer, dapat membantu dalam mengamankan data berupa citra digital.
2. Sebagai bahan referensi dalam pengembangan penelitian lebih lanjut.
3. Sebagai bahan kepustakaan yang dijadikan sarana pengembangan wawasan keilmuan khususnya di bidang matematika terapan pada informatika.

1.5 Batasan Masalah

Agar Pembahasan dalam skripsi ini tidak meluas dan tidak menimbulkan permasalahan yang baru, maka penulis memberi batasan sebagai berikut.

1. Algoritma Super Enkripsi yang digunakan pada penelitian ini adalah Algoritma *Vinegere Cipher* dan *Arnold Cat Map.*
2. Implementasi dari penelitian ini hanya berlaku pada citra berukuran $N \times N$ dengan N merupakan bilangan bulat positif.

1.6 Metode Penelitian

Penulisan dilakukan dengan cara studi literatur. Penulisan dimulai dengan mempelajari jurnal-jurnal, tugas akhir, artikel dan buku-buku tentang enkripsi pada gambar beserta algoritma-algoritmanya. Adapun langkah-langkah untuk menyelesaikan penelitian ini, sebagai berikut.

1. Merumuskan Masalah.
2. Mencari data pendukung secara teoritis.
3. Menyertakan Pesan/ citra (gambar digital).
4. Menyusun enkripsi dengan algoritma Super Enkripsi pada matriks citra.
   a. Menentukan minimal tiga *integer* yang akan digunakan sebagai kunci dan menentukan kunci berupa teks atau simbol.
   b. Merubah kunci teks ke dalam nilai angka dengan bantuan kode ASCII.
   c. Menyusun nilai angka kunci teks dengan setiap entri pada matriks *plain-image* dengan algoritma *Vinegere cipher*.
   d. Mengoperasikan kunci teks dengan setiap entri pada matriks *plain-image* dengan operasi *exclusive-OR.*
   e. Mensubstitusikan tiga integer yang telah dipilih kedalam persamaan *Arnold Cat Map.*
   f. Mentransformasikan posisi tiap entri matriks ke titik lain dengan persamaan *Arnold cat map* sehingga terbentuk *cipher-image.*
5. Menyusun dekripsi algoritma Super Enkripsi pada matriks citra.
   a. Menentukan minimal tiga interger yang akan digunakan sebagai kunci dan menentukan kunci berupa teks atau simbol yang telah digunakan dalam proses enkripsi.

b. Mensubstitusikan tiga integer ke dalam persamaan *Arnold Cat Map.*

c. Mengembalikan posisi entri matriks *cipher-image* ke posisi semula pada *plain-image* dengan melakukan trasformasi titik menggunakan persamaan *Arnold Cat Map.*

d. Merubah kunci teks kedalam nilai angka dengan bantuan kode ASCII.

e. Menyusun nilai angka kunci teks dengan setiap entri pada matriks *plain-image* dengan *Vinegere cipher*.

f. Mengoperasikan nilai angka kunci teks dengan setiap entri pada matriks *plain-image* dengan operasi *exclusive-OR*.

6. Implementasi sebagai ketepatan hasil perhitungan dengan komputer menggunakan aplikasi MATLAB.

7. Intepretasi hasil.

1.7 Sistematika Penulisan

Dalam penulisan dalam penelitian ini, penulis menggunakan sistematika yang terdiri dari empat bab, dan masing-masing bab dibagi dalam subbab dengan sistematika penulisan sebagai berikut.

Bab I Pendahuluan

Membahas tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, tinjauan pustaka, metode penulisan dan sistematika penulisan yang menggambarkan secara singkat isi laporan penelitian ini.

Bab II Kajian Pustaka

Membahas tentang teori-teori penunjang yang digunakan dalam bab selanjutnya, meliputi Matriks, Operasi-operasi matriks, Citra Digital,

Matriks Citra, Enkripsi, dan Dekripsi.

Bab III Pembahasan

Bab ini berisi tentang langkah-langkah pemebentukan Matriks baru yang melalui tahap substitusi dan transposisi yang dilakukan melalui metode *Vinegere cipher* dan *Arnold Cat Map* sehingga didapatkan suatu matriks yang telah terenkripsi dan juga berisi implementasi algoritma kedalam aplikasi MATLAB.

Bab IV Penutup

Bab ini berisi kesimpulan dari pembahasan yang sesuai dengan hasil penelitian, yang selanjutnya dapat digunakan sebagai saran bagi pembaca dan peneliti selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1 Matriks

Definisi

Sebuah matriks adalah susunan segi empat siku-siku dari bilangan-bilangan. Bilangan-bilangan dari susunan tersebut dinamakan entri dalam matriks (Anton, 1997).

Misalkan A merupakan matriks, maka entri-entri dalam matriks A dilambangkan dengan a_{ij} dengan i mewakili entri dalam baris dan j mewakili entri dalam kolom. Maka untuk matriks A berordo $m \times n$ dapat ditulis sebagai berikut dengan $i = 1, 2, \dots n$ dan $j = 1,2, \dots, m$.

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

2.1.1 Operasi-operasi pada matriks

2.1.1.1 Penjumlahan matriks

Definisi

Jika A dan B adalah sebarang dua matriks yang ukurannya sama, maka jumlah $A + B$ adalah matriks yang diperoleh dengan menambahkan bersama-sama entri yang bersesuaian dalam kedua matriks tersebut. Matriks-matriks yang ukurannya berbeda tidak dapat ditambahkan (Anton,1997).

Misalkan $A = [a_{ij}]$, $B = [b_{ij}]$ dengan $i = 1,2,\dots,m$ dan $j = 1,2,\dots,n$

Jika $A + B = C$, maka tiap elemen C akan membentuk

$$C = A + B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

Contoh

Jika A dan B adalah matrik berordo 2×2 dan $C = A + B$ dengan

$$A = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix}, B = \begin{bmatrix} -70 & 121 \\ 7 & -8 \end{bmatrix}$$

Maka

$$C = A + B = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix} + \begin{bmatrix} -70 & 121 \\ 7 & -8 \end{bmatrix} = \begin{bmatrix} 20 + (-70) & 18 + 121 \\ 67 + 7 & 8 + (-8) \end{bmatrix}$$

$$= \begin{bmatrix} -50 & 139 \\ 74 & 0 \end{bmatrix}$$

2.1.1.2 Perkalian matriks

Definisi

Jika *A* adalah suatu matriks dan *c* adalah suatu skalar maka hasil kali (*product*) *cA* adalah matriks yang diperoleh dengan mengalikan masing-masing entri dari *A* oleh *c* (Anton,1997).

Jika c adalah skalar dan A adalah matriks ordo $n \times m$ dengan $i = 1,2,\dots,n$ dan $j = 1,2,\dots,m$

Maka

$$cA = c[a_{ij}] = [ca_{ij}] = \begin{bmatrix} ca_{11} & ca_{12} & \dots & ca_{1n} \\ ca_{21} & ca_{22} & \dots & ca_{2n} \\ \vdots & \vdots & & \vdots \\ ca_{m1} & ca_{m2} & \dots & ca_{mn} \end{bmatrix}$$

Contoh :

Misalkan matriks A berordo 2×2 dan skalar k

$$A = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix}, k = 2$$

Maka

$$kA = 2\begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix} = \begin{bmatrix} 20 \times 2 & 18 \times 2 \\ 67 \times 2 & 8 \times 2 \end{bmatrix} = \begin{bmatrix} 40 & 36 \\ 134 & 16 \end{bmatrix}$$

Definisi

Jika A dalah matriks $m \times r$ dan B adalah matriks $r \times n$, maka hasil kali AB adalah matriks $m \times r$ yang entri-entrinya ditentukan sebagai berikut. Untuk mancari entri dalam baris i dan kolom j dari AB (Anton,1997).

Misalkan matriks $A_{m\times r} = [a_{ij}]$ dan dan matriks $B_{r\times n} = [b_{ij}]$

Jika $A \times B = D$ maka setiap entri D akan membentuk

$$D = A \times B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1r} \\ a_{21} & a_{22} & \dots & a_{2r} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mr} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + \cdots + a_{1r}b_{r1} & a_{11}b_{12} + a_{12}b_{22} + \cdots + a_{1r}b_{r2} & \dots & a_{11}b_{1n} + a_{12}b_{2n} + \cdots + a_{1r}b_{rn} \\ a_{21}b_{11} + a_{22}b_{21} + \cdots + a_{2r}b_{r1} & a_{21}b_{12} + a_{22}b_{22} + \cdots + a_{2r}b_{r2} & \dots & a_{21}b_{1n} + a_{22}b_{2n} + \cdots + a_{2r}b_{rn} \\ \vdots & \vdots & & \vdots \\ a_{m1}b_{11} + a_{m2}b_{21} + \cdots + a_{mr}b_{r1} & a_{m1}b_{12} + a_{m2}b_{22} + \cdots + a_{mr}b_{r2} & \dots & a_{m1}b_{1n} + a_{m2}b_{2n} + \cdots + a_{mr}b_{rn} \end{bmatrix}$$

$$= \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{m1} & d_{m2} & \dots & a_{mn} \end{bmatrix}$$

Contoh.

Jika A merupakan matrik ordo 2×2 dan X merupakan matriks ordo 2×1 dengan

$$A = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Maka jika $D = AX$

$$D = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 20 \cdot 1 + 18 \cdot 3 \\ 67 \cdot 1 + 8 \cdot 3 \end{bmatrix} = \begin{bmatrix} 66 \\ 94 \end{bmatrix}$$

Sehingga

$$\begin{bmatrix} d_{11} \\ d_{21} \end{bmatrix} = \begin{bmatrix} 66 \\ 94 \end{bmatrix}$$

2.1.2 Determinan, Adjoin dan Invers Matriks

2.1.2. 1 Determinan Matriks

Secara umum determinan untuk sebarang matriks persegi berordo $n \times n$ didefinisikan sebagai berikut :

Definisi

Jika A adalah matriks persegi, maka determinan dari matriks A dinotasikan dengan $\det(A)$ atau $|A|$ didefiniskan sebagai jumlah semua hasil kali elementer bertanda dari matriks A (Purwanto, dkk, 2005).

Hasil kali elementer matriks A adalah hasil kali n buah unsur A tanpa ada pengambilan unsur dari baris maupun kolom. Sedangkan hasil kali elementer diberi tanda postif atau negative sehigga dinamakan hasil kali elemneter bertanda tanda negatif atau postif didasarkan pada hasil permutasi .

Contoh :

$$A = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix}$$

Maka

$$\det(A) = \begin{bmatrix} 20 & 18 \\ 67 & 8 \end{bmatrix} = 20 \cdot 8 - 18 \cdot 67 = 160 - 1206 = -1046$$

2.1.2. 2 Adjoin Matriks

Definisi

Jika A adalah matriks $n \times n$, Minor a_{ij} adalah determinan submatrik yang tetap setelah baris ke-i dan kolom ke-j dicoret dari A, dinyatakan dengan $|M_{ij}|$.Sedangkan bilangan $(-1)^{i+j}$ $|M_{ij}|$ dinyatakan oleh C_{ij} disebut Kofaktor (Prakoso, 2014).

Misalkan $A_{n\times n} = [a_{ij}]$

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Maka diperoleh M_{ij} dan C_{ij} dari Matriks A

$$M_{11} = \begin{vmatrix} a_{22} & \dots & a_{2n} \\ \vdots & \dots & \vdots \\ a_{n2} & \dots & a_{nn} \end{vmatrix}, C_{11} = (-1)^{1+1} M_{11}$$

$$\vdots$$

$$M_{nn} = \begin{vmatrix} a_{11} & \dots & a_{1,n-1} \\ \vdots & \dots & \vdots \\ a_{n-1,1} & \dots & a_{n-1,n-1} \end{vmatrix}, C_{nn} = (-1)^{n+n} M_{nn}$$

Jika matriks kofaktor dari A ditranspos maka hasilnya disebut adjont A.

Definisi

Jika A adalah sebarang matriks berordo $n \times n$ dan C_{ij} adalah kofaktor a_{ij}, maka matriks

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

Dinamankan matriks kofaktor dari A. Transpos matriks ini dinamakan adjoin dari A dan dinyatakan dengan $adj(A)$ (Anton dan Rorres, 2010).

Contoh

Misalkan matriks A berordo 3×3

$$A = \begin{bmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{bmatrix}$$

Kofaktor A adalah $c_{11} = 12, c_{12} = 6, c_{13} = -16, c_{21} = 4, c_{22} = 2, c_{23} = 16, c_{31} = 12, c_{32} = -10$ dan $c_{33} = 16$. Matriks kofaktor A adalah

$$\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

Dan adjoin A adalah

$$A = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

2.1.2. 3 Invers Matriks

Definisi

Jika matriks persegi A dikalikan dengan matriks persegi B yang berordo sama, menghasilkan matriks identitas I, yaitu $AB = BA = I$, maka A merupakan invers dari B, atau B merupakan invers dari A. Maka notasi yang digunakan adalah $B = A^{-1}$, sehingga $AA^{-1} = I$ (Andrianto dan Prijo, 2006).

Misalkan A dan B adalah matriks berordo $n \times n$

$$A_{n\times n} = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, B_{n\times n} = [b_{ij}] = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}$$

Jika

$$AB = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$BA = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Maka

$$A^{-1} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \text{ dan } B^{-1} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Contoh

$$A = \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} dan\ B = \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}$$

Maka

$$AB = \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$BA = \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Untuk mencari invers matriks dapat menggunakan determinan dan adjoin sebagai berikut

Teorema

Jika matriks A dapat dibalik jika dan hanya jika $\det(A) \neq 0$, maka

$$A^{-1} = \frac{1}{\det(A)} adj\ (A)$$

Bukti

Misalkan $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$ dengan $\det(A) \neq 0$.

$$Aadj(A) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{m1} & C_{m2} & \dots & C_{mn} \end{bmatrix},$$ hasil kali matriks A dengan $Adj(A)$ yaitu, baris pertama kolom pertama hasil kali adalah $a_{11}C_{11} + a_{12}C_{12} + a_{13}C_{13} + \cdots + a_{1n}C_{1n}$, baris pertama kolom kedua hasil kali adalah $a_{11}C_{21} + a_{21}C_{22} + a_{13}C_{23} + \cdots + a_{1n}C_{2n}$, dan seterusnya. Secara umum hasil kali matriks A dengan $Adj(A)$ baris ke i kolom ke j adalah $a_{j1}C_{j1} + a_{i2}C_{j2} + \cdots + a_{in}C_{jn}$. Ambil hasil kali pada diagonal utama yaitu $i = j$, maka diperoleh $a_{i1}C_{j1} + a_{i2}C_{j2} + a_{i3}C_{j3} + \cdots + a_{in}C_{jn} = \det(A)$. sebaliknya hasil kali selai diagonal utama yitu $i \neq j$, maka entri a dan kofaktor-kofaktornya berasal dari matriks A yang berbeda , sehingga hasilnya adalah 0. Diperoleh hasil kali matriks A yang berbeda, sehingga hasilnya adalah 0. Diperoleh hasil kali matriks A dengan$adj(A)$ yaitu,

$$Aadj(A) = \begin{bmatrix} |A| & 0 & \dots & 0 \\ 0 & |A| & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & |A| \end{bmatrix} = |A| \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \det(A)I$$

Diperoleh $Aadj(A) = \det(A)I$ atau $\frac{Aadj(A)}{\det(A)} = I$, kemudian dikalikan dengan A^{-1} menjadi $A^{-1}\frac{Aadj(A)}{\det(A)} = A^{-1}I$ atau $AA^{-1}\ \frac{1}{\det(A)}adj\ (A) = A^{-1}I$, karena $AA^{-1} = I$ dan $A^{-1}I = A^{-1}$ maka diperoleh $\frac{1}{\det(A)}adj(A) = A^{-1}$ atau $A^{-1} = \frac{1}{\det(A)}\ adj\ (A)$.

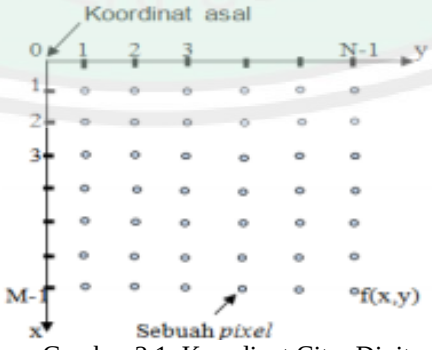
2.2 Citra digital

Citra (*image*) adalah istilah lain untuk gambar. Sebagai salah satu komponen multimedia yang memegang peranan sangat penting dalam bentuk informasi visual, citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu kaya dengan informasi (Sutoyo, dkk., 2009).

Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut pencitraan (*imaging*) atau digitalisasi. Untuk mengubah bentuk kontinu ke bentuk digital, memerlukan dua fungsi yaitu koordinat dan tingkat keabuhan *(gray-level)*. Dikritisasi pada nilai koordinat disebut spasial (*sampling*) sedangkan pada tingkat keabuhan*(gray-level)* disebut kwantisasi (*quantization)* (Gonzales, 2002).

2.2.1 Matriks Citra Digital

Suatu citra dapat didefinisikan sebagai fungsi $f(x, y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut (Nafi'iyah,2015). Gambar 2.1 menunjukkan posisi koordinat citra digital.



Gambar 2.1. Koordinat Citra Digit

Notasi Gambar 2.1 memungkinkan untuk ditulis dalam citra digital $M \times N$ sebagai berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Pada sisi kanan dari persamaan adalah definisi dari citra digital. Setiap elemen dari matrik disebut dengan element citra, elemen gambar, atau *pixel*. Beberapa notasi matriks menotasikan citra digital dan elemennya sebagai berikut.

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Secara jelas , $a_{ij} = f(x = i, y = j) = f(i,j)$ jadi persamaan dari matriks $f(x,y)$ dan matriks A merupakan matriks identik.

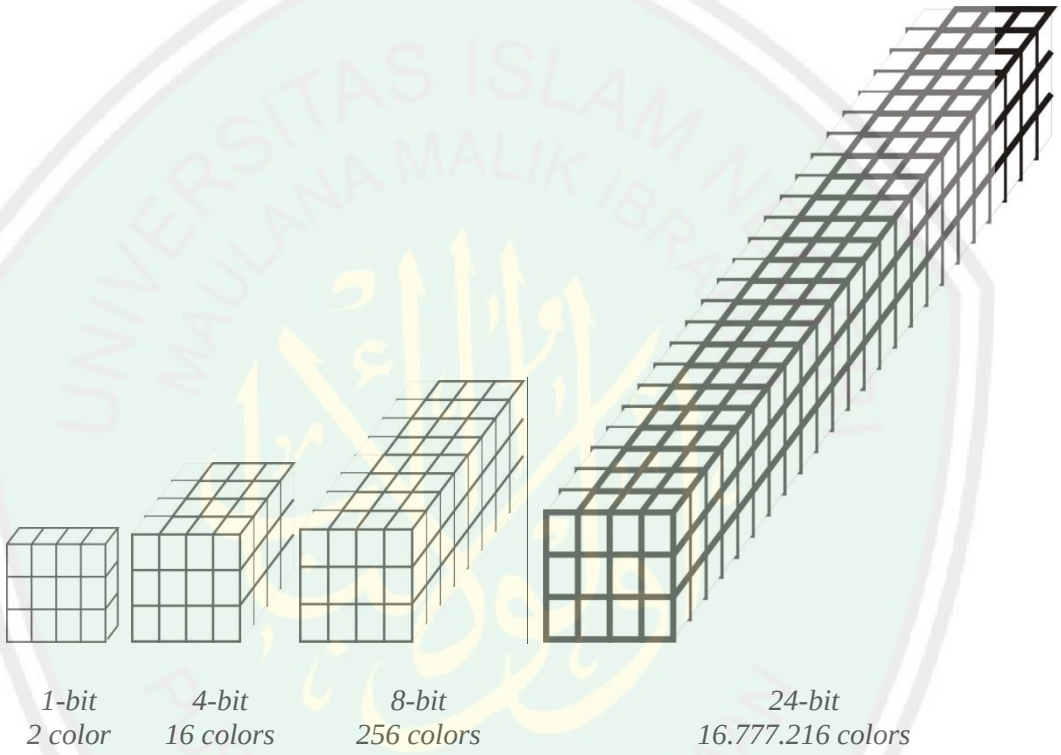
Untuk mempermudah penggambaran matriks citra dapat digunakan kumpulan persegi yang akan mewakili setiap entri dalam matriks citra seperti pada gambar 2.2.

Gambar 2.2 . Koordinat Citra berukuran 4×4 dalam Visualisasi Persegi

2.2.2 Jenis Citra dan Komposisi Warna

Dalam sebuah citra terdapat komposisi antara koordinat dan tingkat keabuhan (*Gray-Level),* pada 2.2 koordinat diartikan sebagai posisi entri dalam matrik citra maka tingkat keabuhan merupakan entri/ nilai dari setiap entri/pixel

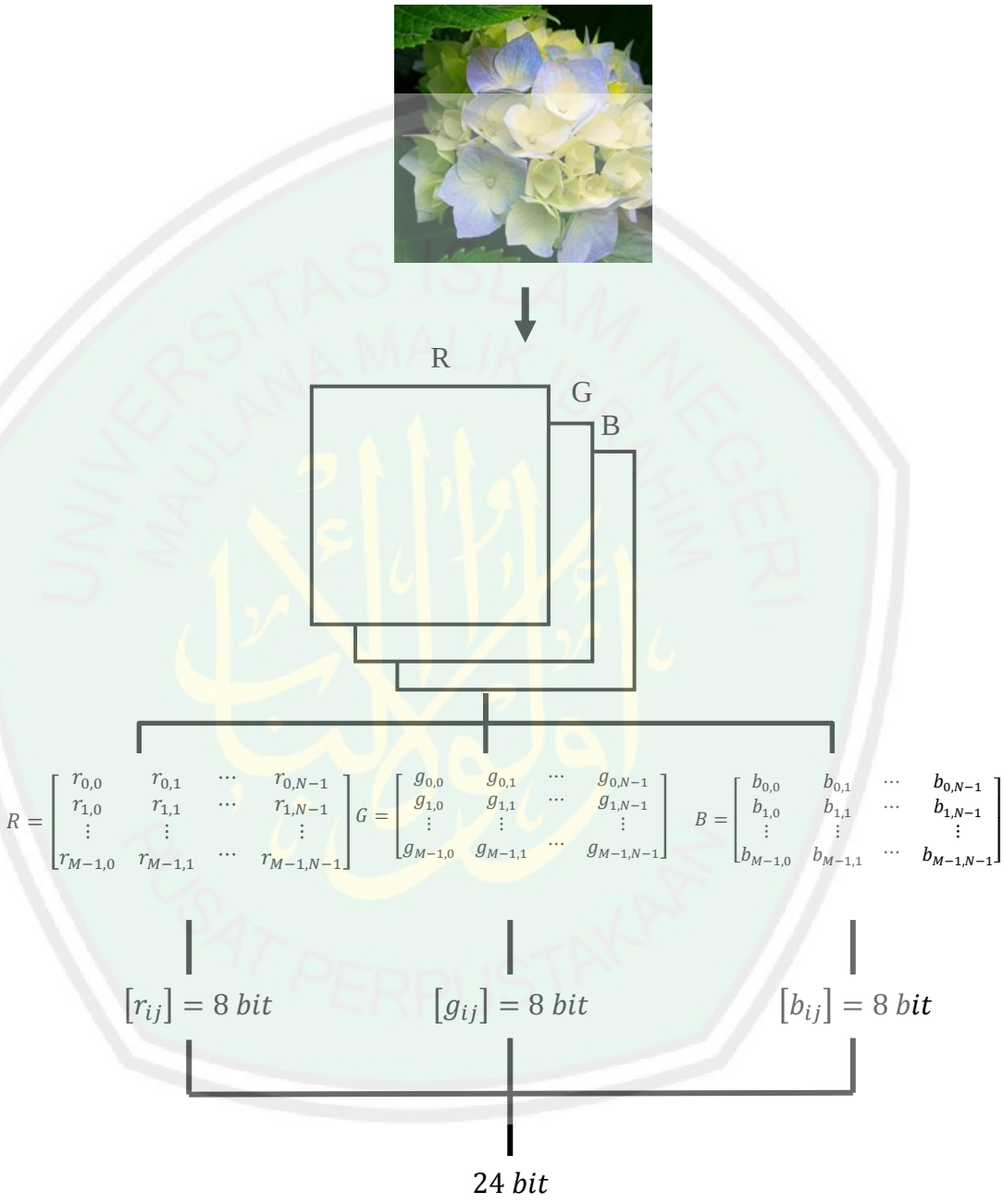
citra yang akan menentukan warna pada tiap pixel. Tingkat keabuhan ini berkaitan dengan binary digital (bit) yang tersusun atas bilangan 0 dan 1. Semua varian warna untuk pixel diperoleh dari tiga warna dasar yaitu merah, hijau dan biru. Setiap warna dasar dipresentasikan dengan 1 byte; gambar 24 bit menggunakan 3 byte per pixel untuk mempresentasikan suatu nilai (Ariyus, 2006). Hubungan antara koordinat citra dengan *gray -level* dapat dilihat pada gambar berikut..



Gambar 2.3 Visualisasi *Gray-level* dalam Citra

Pada gambar 2.3 masing-masing kubus mempresentasikan data dengan citra yang berukuran 4×4 pixel dengan bentuk kubus memanjang ke belakang yang mempresentasikan jumlah bit yang digunakan dalam tiap pixel. kedalaman 8-bit atau kurang memepresentasikan gambar *grayscale* sedangkan citra 24-bit mengandung warna RGB (*Red Green Blue)*(Binanto, 2010).

Berikut ini adalah penggambaran matriks citra yang memiliki warna RGB. Disebut sebagai citra RGB dikarenakan dalam citra tersbeut ada tiga layer yang akan menciptakan lebih dari 16 ribu jenis warna.



Gambar 2.4 Matriks dalam Citra RGB

Setiap entri dalam matrik R, G atau B memiliki rentan nilai [0,255], dimana tiap nilai jika dirubah kebentuk biner memiliki nilai tersusun atas 8 bit atau 8 susunan angka biner. Tiap entri tersebut dalam rentan nilai [0,255] memiliki *gray lavel* tersendiri. Sedangkan pada gambar *grayscale* hanya memiliki satu layer, sehingga warna yang dihasilkan hanya warna *gray* karena tidak ada susunan layer lain yang akan menciptakan konbinasi warna yang lebih banyak.

2.3 Kriptografi

Kriptografi atau yang sering dikenal dengan sebutan ilmu penyandian data adalah suatu bidang ilmu dan seni (*art and science*) yang bertujuan untuk menjaga kerahasiaan suatu pesan yang berupa data-data (Sutoyo, dkk., 2009). Menjaga kerahasian pesan dilakukan untuk menjaga privasi kelompok atau perseorangan. Berikut ini beberapa contoh dalam kehidupan dimana pesan/data memiliki sifat rahasia yang hanya diperuntukan untuk kelompok/ perseorangan (McAndrew, 2011).

1. Rekening bank dan dokumen keuangan. Jika nasabah mengirim pesan permintaan menarikan uang ke bank, dan kemungkinan dalam pesan tersebut terdapat data diri nasabah, dan informasi tentang akun bank. Dalam hal ini nasabah menginginkan datanya untuk dirahasiakan dari umum.
2. Nomor kartu kredit. Membeli secara online telah menjadi hal yang biasa, namun pembeli menginginkan detail dari kartu kredit hanya bisa diakses oleh pihak penjual.
3. Rekam medis. Dalam peraturan, rekam medis adalah data rahasia, dan hanya boleh diakses oleh pasien, tenaga medis dan orang yang diizinkan.

4. Perintah militer. Dalam perintah untuk mencapai suatu misi, perintah militer biasanya disampaikan dalam kode.
5. Informasi pribadi. Beberapa orang biasanya lebih memilih menutupi informasi pribadinya dari akeses umum.

Kriptografi dalam kacamata modern dapat dianggap sebagai pengaplikasian operasian struktur matematika tertentu (Sadikin, 2012). Ada 3 fungsi dasar dalam kripsografi, yaitu:

1. Enkripsi : proses pengamanan terhadap data/pesan.
2. Dekripsi : proses kebalikan dari enkripsi.
3. *Key* : kunci yang digunakan dalam proses enkripsi dan dekripsi.Umumnya algoritma enkripsi dan dekripsi bergantung pada kunci (*key*) rahasia. Kunci rahasia biasanya ini tersusun angka, alphabet, simbol maupaun barisan bit-bit.

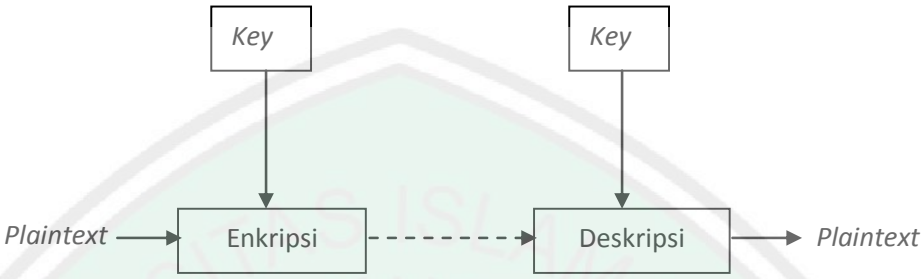
2.3. 1 Enkripsi

Jika ditinjau dari pengertiannya maka algoritma Enkripsi merupakan langkah-langkah sistematis yang digunakan dalam menyandikan/ menyembunyikan pesan dari pihak-pihak lain yang tidak berhak atas pesan tersebut. Keamanan dari algoritma enkripsi tergantung dari bagaimana suatu algoritma itu bekerja, maka algoritma semacam ini disebut dengan algoritma terbatas (Ariyus, 2006).

Berdasarkan kunci yang dipakai, algoritma enkripsi memiliki tiga macam bentuk (Ariyus, 2006):

1. Algoritma Simetri

Algoritma ini sudah ada lebih dari 4000 tahun yang lalu. Algoritma ini juga sering disebut dengan algoritma klasik, karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsinya (Ariyus, 2006).

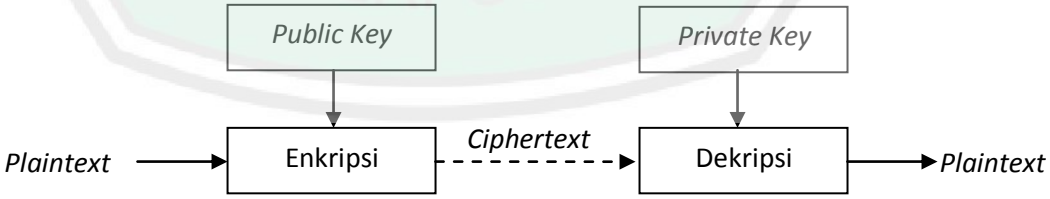


Gambar 2.5 Sistem *Key secret* dalam enkripsi dan dekripsi

2. Algoritma Asimetri

Algoritma asimetri sering disebut juga dengan lagoritma fungsi public. *Public-Key* (kunci publik) merupakan penggunaan kunci yang berbeda dalam proses enkripsi maupun dekripsinya (McAndrew, 2011). Pada algoritma asimetri kunci terbagi menjadi dua bagian (Ariyus, 2006):

- Kunci umum (*public key*) : kunci yang boleh semua orang tahu (dipublikasikan).
- Kunci pribadi (*private key*) : kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).



Gambar 2.6 Sistem *Public Key* dalam enkripsi dan dekripsi

3. Hash Function

Fungsi hash sering disebut dengan fungsi satu arah (*one-way function*), *massage digest*, *fingerprint*, fungsi kompre dan *massage authentication code*

(MAC). Hal ini merupakan fungsi matematika yang mengambil input panjang variabel dan mengubahnya ke dalam urutan binner dengan panjang yang tetap. Fungsi hash biasanya diperlukan bila ingin membuat sidik jari dari suatu pesan.

2.3. 2 Dekripsi

Dalam proses dekripsi *ciphertext* akan dirubah menjadi *plaintext* kembali dengan menggunakan kunci yang juga digunakan dalam enkripsi (Stalling, 2003). Pada algoritma Simetri kunci pada enkripsi dan dekripsi adalah sama, sedangkan dekripsi pada pada algoritma Asimetri kunci yang digunakan berbeda dengan enkripsi atau disebut dengan kunci public. Algoritma yang memakai kunci publik diantaranya adalah (Ariyus, 2006):

1. *Digital signature algoritm (DSA)*
2. *RSA*
3. *Diffie-Hellman (DH)*
4. *Elliptic Curve Cryptography (ECC)*
5. Dan lain sebagainya.

Enkripsi dan dekripsi merupakan suatu proses yang tidak dapat dipisahkan, jika suatu metode enkripsi tidak memiliki pemecahannya atau dekripsinya dapat dikatakan bahwa metode tersebut *fail*. Secara singkat dekripsi dapat dinyatakan sebagai proses kebalikan dari enkripsi.

2.4 Super Enkripsi

pada dasarnya algoritma kriptografi klasik dibagi menjadi dua yaitu, subsitusi dan permutasi yang telah di bahas sebelumnya. Kedua teknik ini termasuk algoritma yang mudah di pecakan melalui *brute force*. *Brute force* adalah memecahkan sandi dengan mencoba seluruh sandi yang mungkin ke dalam

ciphertext sampai dapat dirubah ke dalam *plaintext*. Rata-rata, setengah dari seluruh kemungkinan kunci harus dicoba agar dapat berhasil (Stallings, 2003).

Untuk mempersulit algoritma sehingga kunci tidak mudah di temukan oleh pihak yang tidak bersangkutan, maka dikembangkan algoritma baru dengan menggabungkan kedua teknik algoritma klasik tersebut. Super enkripsi merupakan suatu konsep dengan menggunakan kombinasi dari dua atau lebih dari teknik subsitusi dan transposisi *cipher* untuk mendapatkan suatu algoritma yang lebih andal (susah di pecahkan)(Ariyus, 2006). Untuk menjalankan teknik super enkripsi ini, harus memahami teknik subsitusi dan transposisi. Super enkripsi dijalankan dengan melakukan enkripsi pesan dengan teknik subsitusi, selanjutnya *ciphertext* yang telah didapatkan dienkripsi lagi dengan teknik transposisi.

2.5 *Vinegere Cipher*

Cipher ini melambangkan Blaise Vinegere (1532-1596), yang merupakan diplomat Vatican (Stallings, 2011). Meskipun pada kenyataannya bukan ia yang menciptakan *cipher* ini. *Vinegere cipher* diciptakan oleh Giovan Battista Bellaso di tahun 1533. Walaupun sejarah telah mencatat nama Giovan, *cipher* ini pada akhirnya lebih dikenal dengan *Vinegere cipher*.

Teknik dari subsitusi *Vinegere* bisa dilakukan dengan dua cara, yaitu dengan huruf dan angka (Ariyus, 2006). Berikut ini *Vinegere cipher* dengan metode angka.

Tabel 2.1 Pertukaran huruf dengan bilangan

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Suatu kata "*cipher*" akan digunakan sebagai kunci dan "rahasiaku" akan dienkripsi dengan kunci tersebut. Dari kata "*cipher*" kita memiliki kunci dalam angka K= (2, 8, 15, 7, 4, 17). Dan dalam kata "rahasiaku" diperoleh angka P= (17, 0, 7, 0, 18, 8, 0, 10, 20). Kedua himpunan angka ini akan dikombinasikan dengan menggunakan Algoritma enkripsi dirumuskan sebagai berikut (Stalling, 2011).

$$c_i = p_i + k_i \ (mod\ 26)$$

Dimana p_i, k_i dan c_i berturut turut adalah variabel yang mewakili tiap karakter dalam *plaintext, keyword* dan *ciphertext.* $(mod\ 26)$ dilakukan untuk membatasi nilai agar tidak keluar dari jumlah *alphabet.*

2.6 *Arnold Cat Map*

Metode *Arnold Cat Map* (ACM) diperkenalkan pertama kali oleh seorang ahli matematik Rusia yang bernama Vladimir I. Arnold, pada tahun 1960 yang mendemonstrasikan algoritmanya tersebut dengan menggunakan citra kucing (Purba, 2014).

Metode ACM ini akan mentrasformasikan titik asal pixel yang dipresentasikan dengan (x, y) menjadi (x', y') yang merupakan hasil transformasi dari titik asal. Maka semua titik pixel akan teracak sesuai iterasi yang diberikan. Metode *Arnold Cat Map* dirumuskan sebagai berikut :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N)$$

Dimana (x', y') merupakan hasil tranformasi titik asal (x, y), sedangkan a, b nilai integer yang bernilai sembarang. Determinan matriks harus sama dengan

1 agar hasil transformasinya bersifat satu-satu, sehingga setiap titik akan ditransformasikan kehanya satu titik. Sedangkan N merupakan lebar/panjang citra, hasil dari perkalian matriks akan dimodulokan dengan panjang citra agar hasilnya *area-preserving*, yaitu tetap berada di dalam area citra yang sama (Munir, 2012).

Karena bentuk matriks telah didefinisikan sebagai $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ maka berapun nilai integer positif yang mewaili b dan c , matriks akan memiliki determinan 1. ACM diiterasikan sebanyak m kali , setiap iterasi menghasilkan citra acak. Nilai b, c dan m dianggap sebagai kunci rahasia dalam kriptografi simetri.

2.7 *Bit-String* dalam Kriptografi Modern

Kriptografi modern berbeda dengan kriptografi klasik dikarenakan pada pengoperasiannya sudah menggunakan computer, yang berfungsi mengamankan data baik yang ditransfer melalui jaringan computer maupun tidak, hal ini sangat berguna untuk melindungi privasi, integritas data (Ariyus, 2006).

Pada kriptografi klasik, menggunakan teknik subsitusi dan transposisi karakter dari *plaintext* , dan hasil dari subsitusi dan tranposisi akan menghasilkan *ciphertext.* Pada kriptografi modern karakter yang ada dikonversi kedalam suatu urutan digit biner (bits) yaitu 1 dan 0, yang umum digunakan untuk schema encoding ASCII (*American Standart Code for Information Interchange)*. *Squence bit* (urutan bit) yang akan mewakili *plaintext* yang kemudian akan dienkripsi untuk mendapatkan *chipertext* dalam bentuk *sequence bit*.

Algoritma enkripsi bisa menggunakan salah satu dari dua metode, metode yang pertama "natural" pembagian antara *stream cipher*, dimana urutan

bit untuk enkripsi digunakan *bit by bit*. Metode kedua adlah *block cipher*, dimana urutan pembagian dalam bentuk ukuran block yang dinginkan. ASCII memerlukan 8 bit untuk emndapatkan satu karakter dan block cipher mempunyai 64 bit untuk satu block. Sebagai contoh sequence 12 bit : 100111010110, dipecah menjadi 3 block maka akan di dapatkan 100 111 010 110. Bagaimanapun, Bit-String dengan panjang 3 menghadirkan bilangan bulat 0 sampai 7 dengan urutan menjadi 4 7 2 6 (Ariyus, 2006).

$$000 = 0, \quad 001 = 1, \quad 010 = 2, \quad 011 = 3, \quad 100 = 4,$$

$$101 = 5, \quad 110 = 6, \quad 111 = 7$$

Sejak operasi algoritma *cipher* menggunakan binari string , maka perluh dibiasakan menggunakan metode kombinasi dua bit yang disebut dengan *Exclusive OR* atau disebut XOR yang ditandai dengan $\oplus$. Ini merupakan suatu penambahan modulo 2 dan digambarkan sebagai beriku $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$. Operasi XOR ini mengkombinasikan dua bit-string dengan panjang yang sama (Ariyus, 2006).

BAB III

PEMBAHASAN

3.1 Analisa Enkripsi *Vinegere cipher* dan *Arnold Cat Map*

3.1.1 Algoritma Enkripsi dengan *Vinegere cipher*

Dalam penelitian ini, algoritma yang digunakan adalah *Vinegere cipher* dengan modifikasi XOR. Jika pada versi asli algoritma ini diimplementasikan dengan menggunakan operasi tambah "+", pada penelitian ini algoritma tersebut akan diimplementasikan dengan menggunakan *exclusive-OR* (XOR). XOR adalah operasi biner yang sering digunakan dalam *cipher* yang beroperasi dalam mode bit. Notasi matematis untuk operator XOR adalah $\oplus$, selain itu Operasi XOR juga identik dengan penjumlahan pada modulo 2. Berikut aturan bit yang dioperasikan dengan XOR

Tabel 3.1 Operasi XOR

a	b	$a \oplus b$	Penjumlahan dalam modulo 2
0	0	0	$0 + 0\ (mod 2) = 0$
0	1	1	$0 + 1\ (mod\ 2) = 1$
1	0	1	$1 + 0\ (mod\ 2) = 1$
1	1	0	$1 + 1 (mod\ 2) = 0$

Maka rumus umum dari *Vinegere cipher* dimodifikasi sedemikian rupa sehingga cocok digunakan dalam kriptografi citra digital.

Misalkan P merupakan matriks dari *plain-image* , C matriks *cipher-image.* Misalkan matriks P dan C memiliki ordo $N \times N$, Matriks-matriks tersebut kemudian disusun menjadi array dimensi 1 dengan mengambil baris ke-1 sampai

ke-N. Jika $q \in P$ dan $r \in C$ maka array dari P dan C dapat disusun sebagai berikut.

$$\{\underbrace{q_{1,1}, q_{1,2}, \dots, q_{1,N}}_{q_{1,n}}, \underbrace{q_{2,1}, q_{2,2}, \dots, q_{2,N}}_{q_{2,n}}, \dots, \underbrace{q_{N,1}, q_{N,2}, \dots, q_{N,N}}_{q_{N,n}}\}$$

Dengan $n = 1,2, \dots, N$. Begitu pula dengan penyusunan array pada matriks C

$$\{r_{1,1}, r_{1,2}, \dots, r_{1,N}, r_{2,1}, r_{2,2}, \dots, r_{2,N}, \dots, r_{N,1}, r_{N,2}, \dots, r_{N,N}\}$$

Maka jika dituliskan kembali entri di dalam P_1 dan C_1 yang berturut-turut merupakan array dimensi satu dari P dan C akan memiliki panjang N^2sebagai berikut.

$$P_1 = \{p_1, p_2, p_3, \dots, p_{N^2}\}$$

$$C_1 = \{c_1, c_2, c_3, \dots, c_{N^2}\}$$

Jika $p_i \in P_1$, $c_i \in C_1$ dan $k_j \in K$ yang merupakan himpunan kunci dengan $1 \leq i \leq N^2$ serta $1 \leq j \leq t$ dengan t merupakan panjang kunci. Maka persamaan *Vinegere cipher* modifikasi *XOR* dapat dirumusakan sebagai berikut.

$$\boldsymbol{c_i = p_i \oplus k_j} \tag{3.1}$$

Keterangan :

$c = nilai\ pixel\ pada\ cipher\ image$

$p = nilai\ pixel\ pada\ plain - image$

$k = nilai\ kunci\ dalam\ desimal$

$\oplus = notasi\ operasi\ Exclusive - OR$

$i = 1,2,,3, \dots, N^2$

$j = 1,2,3, \dots, t$

$N = ordo\ panjang/lebar\ matriks$

$t = panjang\ kunci$

Berikut ini algoritma dalam melakukan enkripsi dengan menggunakan algoritma *Vinegere cipher* modifikasi *XOR.*

1. Memasukkan teks sebagai kunci algoritma *Vinegere cipher*, minimal teks terdiri dari 3 karakter. Serta menyertakan *plain-image* (citra asli) berukuran $N \times N$ pixel yang akan dienkripsi.
2. Membaca matriks *plain-image* sebagai matriks $N \times N$. *Plain-image* (citra asli) akan dibagi menjadi 3 layer yaitu layer R*(Red), G(green)* dan B *(blue).* Ketiga layer tersebut akan menciptakan tiga matriks yang berbeda namun memiliki ordo yang sama dengan *plain-image.*
3. Membentuk matriks *plain-image* sebagai array *plain-image* dimensi 1.
4. Teks akan diterjemahkan kedalam angka dengan kode ASCII, sehingga diperoleh $k_1, k_2, \dots, k_t$ dengan t= panjang kunci/karakter.
5. Membentuk kunci $k_1, k_2, \dots, k_t$ sebagai array sepanjang array *plain-image.*
6. Setiap kunci $k_1, k_2, \dots, k_t$ yang berbentuk decimal akan dirubah dalam bentuk *binary.* Begitu pula pada *plain-image*, setiap nilai pixel yang berbentuk desimal dirubah dalam bentuk *binary*.
7. Menjalankan algoritma *Vinegere cipher* dengan mengoperasikan array *plain-image* dan array kunci menggunakan operasi XOR dengan menggunakan persamaan (3.1) sehingga membentuk array dengan entri baru yang dimisalkan sebagai array *cipher*.
8. Array *cipher* akan diubah menjadi matriks *cipher*, Matriks *cipher* ini kemudian akan membentuk citra *chiper-image.* Enkripsi dilakukan pada tiga matriks citra baik layer *Red (R),* layer *Green(G)* dan layer *blue(B)* melakukan

langkah 2 sampai 8 secara terpisah. Kemudian *cipher-image* akan terbentuk dengan menggabungkan ketiga matriks tersebut secara tertumpuk berturut-turut dari matriks *cipher* dari layer R, G dan B.

Berikut ini penerapan algoritma *Vinegere cipher* modifikasi XOR pada *plain-image* berukuran 3×3 pixel. Penerapan dilakukan pada *plain-image grayscale*, sebuah citra *grayscale* dapat tersusun dengan satu layer saja. Walaupun pada umumnya semua citra digital baik *grayscale* atau berwarna tersusun atas layer RGB. Pada contoh berikut ini digunakan citra *grayscale* yang tersusun satu layer.

1. *plain-image grayscale* berukuran 3×3 pixel akan di enkripsi dengan kunci "Key". Misalkan *plain-image* sebagai berikut.

Gambar 3.1 . *Plain-image grayscale* ukuran 3×3

2. Selanjutnya *plain-image* berukuran 3×3 pixel dibaca sebagai matrik berordo 3×3 sebagi berikut.

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

3. Kunci "Key" diterjemahkan dalam desimal dengan bantuan kode ASCII didapatkan $k_1 = 75,\ k_2 = 101,\ k_3 = 122.$
4. Membentuk array dari matriks *plain-image* sehingga diperoleh

$$P_1 = \{140,\ 146, 163,\ 124,\ 122,\ 152,\ 114,\ 95,\ 116\}$$

5. Membentuk array dengan entri $k_1, k_2, dan\ k_3$ dengan mengulang entri tersebut sepanjang P_1. Maka

$$K = \{k_1, k_2, k_3, k_1, k_2, k_3, k_1, k_2, k_3\}$$

Sehingga

$$K = \{\ 75, 101, 122, 75, 101, 122, 75, 101, 122\ \}$$

6. Jika setiap entri dalam array P dan K dirubah kedalam bentuk angka binary maka diperoleh

$$p_1 = 140_{10} = 10001100_2$$
$$p_2 = 146_{10} = 10010010_2$$
$$p_3 = 163_{10} = 10100011_2$$
$$p_4 = 124_{10} = 01111100_2$$
$$p_5 = 122_{10} = 01111010_2$$
$$p_6 = 152_{10} = 10011000_2$$
$$p_7 = 114_{10} = 01110010_2$$
$$p_8 = 95_{10} = 01011111_2$$
$$p_9 = 116_{10} = 01110100_2$$

Serta nilai entri K sebagai berikut.

$$k_1 = 75_{10} = 00101011_2$$
$$k_2 = 101_{10} = 01100101_2$$
$$k_3 = 122_{10} = 01111010_2$$

7. Dengan menggunakan persamaan (3.1) diperoleh hasil berikut.

$$c_1 = p_1 \oplus k_1 = 140_{10} \oplus 75_{10} = 10001100_2 \oplus 00101011_2 = 10100111_2$$
$$= 167_{10}$$

$$c_2 = p_2 \oplus k_2 = 146_{10} \oplus 101_{10} = 10010010_2 \oplus 01100101_2 = 1111011_2$$
$$= 247_2$$

$$c_3 = p_3 \oplus k_3 = 163_{10} \oplus 122_{10} = 10100011_2 \oplus 01111010_2 = 11011001_2$$
$$= 217_{10}$$

$$c_4 = p_4 \oplus k_1 = 124_{10} \oplus 75_{10} = 01111100_2\ 00101011_2 = 01010111_2$$
$$= 87_{10}$$

$$c_5 = p_5 \oplus k_2 = 122_{10} \oplus 101_{10} = 01111010_2 \oplus 0110010_2 = 00011111_2$$
$$= 31_{10}$$

$$c_6 = p_6 \oplus k_3 = 152_{10} \oplus 122_{10} = 10011000_2 \oplus 01111010_2 = 11100010_2$$
$$= 226_{10}$$

$$c_7 = p_7 \oplus k_1 = 114_{10} \oplus 75_{10} = 01110010_2 \oplus 00101011_2 = 0101100_2$$
$$= 89_{10}$$

$$c_8 = p_8 \oplus k_2 = 95_{10} \oplus 101_{10} = 01011111_2 \oplus 01100101_2 = 00111010_2$$
$$= 58_{10}$$

$$c_9 = p_9 \oplus k_3 = 116_{10} \oplus 122_{10} = 01110100_2 \oplus 01111010_2 = 00001110_2$$
$$= 28_{10}$$

Dari hasil subsitusi diatas diperoleh C_1

$$C_1 = \{167,247,217,87,31,226,89,58,28\}$$

8. C_1akan dirubah kedalam bentuk matriks kembali dengan ordo yang sama dengan matriks *plain-image*.

$$C = \begin{bmatrix} 167 & 247 & 217 \\ 87 & 31 & 226 \\ 89 & 58 & 28 \end{bmatrix}$$

Matrik C inilah yang akan menjadi dasar pembentukan citra baru yang telah dienkripsi yang disebut dengan *cipher-image*.

Gambar 3.2 *Cipher-image Grayscale* enkripsi *Vinegere cipher*

3.1.2 Algoritma Enkripsi dengan *Arnold Cat Map*

Metode *Arnold Cat Map* pada dasarnya memiliki konsep yang sama dengan tranformasi geometri dalam matematika. Metode ini akan mentransformasikan titik (x, y) di *plain-image* pada titik lain (x', y'), sehingga hasil trasnformasi dari seluruh titik yang ada pada *plain-image* selanjutnya hasil tranformasi seluruh titik pixel ini akan disebut sebagai *cipher-image*. Berikut ini adalah persamaan yang digunakan untuk mentrasformasikan titik pixel.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N) \tag{3.2}$$

Keterangan :

$x = titik\ awal\ pixel\ di\ koordinat\ x$; $x = 0, 1, \dots, N-1$

$y = titik\ awal\ \ pixel\ di\ koordinat\ y$; $y = 0,1, \dots, N-1$

$x' = titik\ pixel\ hasil\ \ tranformasi\ di\ koordinat\ x$

$y' = titik\ pixel\ hasil\ transformasi\ di\ koordinat\ y$

$a, b = bilangan\ bulat\ positif$

$N = \ ordo\ matriks\ citra\ dalam\ kolom\ \ / baris$

Berikut ini algoritma dalam melakukan enkripsi dengan menggunakan algoritma *Arnold cat map.*

1. Memasukkan kunci berupa tiga integer (a, b, d)Serta menyertakan *plain-image* (citra asli) berukuran $N \times N$ pixel dengan $N \in$ integer positif yang akan dienkrispi.
2. Membaca *plain-image* berukuran $N \times N$ sebagai matriks $N \times N$. *Plain-image*(citra asli) dibagi menjadi layer R*(Red), G(green)* dan B *(blue)*. Setiap layer tersusun atas matriks, maka ada tiga matriks yang terbaca sebagai matriks *plain-image*.
3. Masukkan integer *a* dan *b* dalam persaman (*3.2)*.
4. Selanjutnya dari matriks $N \times N$ akan diambil titik koordinat setiap entri, titik koordinat tersebut akan di kalikan dengan matrik 2×2 yang entrinya merupakan *integer a* dan *b*. Proses transformasi ini menggunakan persamaan (3.2).
5. Transformasi dilakukan terus menerus sampai iterasi *d* dan seluruh pixel dalam terpetakan dan menghasilkan matriks baru. Enkripsi dilakukan pada tiga matriks citra baik matrik dari layer *Red (R),* layer *Green(G)* dan layer *blue(B)* melakukan langkah 3 dan 5 secara terpisah. Kemudian *cipher-image* akan terbentuk dengan menggabungkan ketiga matriks tersebut secara tertumpuk berturut-turut dari matriks *cipher* dari layer R, G dan B.yang disebut *cipher-image*.

Berikut ini penerapan algoritma *Arnold cat Map* pada *plain-image* berukuran 3×3 pixel. Penerapan dilakukan pada *plain-image grayscale*, sebuah citra *grayscale* dapat tersusun dengan satu layer saja. Walaupun pada umumnya semua citra digital baik *grayscale* atau berwarna tersusun atas layer RGB. Pada contoh berikut ini digunakan citra *grayscale* yang tersusun satu layer.

Misalkan kunci yang digunakan adalah $= 2$,$b = 4$ dan $d = 1$. Dengan *plain-image* berukuran 3×3 sebagai berikut.

Gambar 3.3 *plain-image grayscale* berukuran 3×3

1. Selanjutnya *plain-image* berukuran 3×3 akan dibaca sebagai matriks 3×3.

   Misalkan matriks mempunyai entri sebagai berikut.

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

2. Memasukkan kunci ke dalam persamaan (3.2).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N)$$

3. Berikut proses transformasi titik matrik P pada matrik C.

   jika $p_{x,y} \epsilon P$ dan $c_{x,y} \in C$ dengan titik $x = 0,1,2$ dan $y = 0,1,2$

   untuk baris $x = 0$ dan kolom $y = 0$, dengan entri $p_{0,0} = 140$

$$p_{0,0} = 140 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow c_{0,0}$$

   untuk baris $x = 1$ dan kolom $y = 0$, dengan entri $p_{1,0} = 124$

$$p_{1,0} = 124 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 4 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow c_{1,1}$$

   untuk baris $x = 2$ dan kolom $y = 0$, dengan entri $p_{2,0} = 114$

$$p_{2,0} = 114 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 8 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow c_{2,2}$$

   untuk baris $x = 0$ dan kolom $y = 1$, dengan entri $p_{0,1} = 146$

$$p_{0,1} = 146 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 9 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \rightarrow c_{2,0}$$

untuk baris $x = 1$ dan kolom $y = 1$, dengan entri $p_{1,1} = 122$

$$p_{1,1} = 122 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 3 \\ 13 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow c_{0,1}$$

untuk baris $x = 2$ dan kolom $y = 1$, dengan entri $p_{2,1} = 95$

$$p_{2,1} = 95 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 4 \\ 17 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow c_{1,2}$$

untuk baris $x = 0$ dan kolom $y = 2$, dengan entri $p_{0,2} = 163$

$$p_{0,2} = 163 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 4 \\ 9 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow c_{1,0}$$

untuk baris $x = 1$ dan kolom $y = 2$, dengan entri $p_{1,2} = 146$

$$p_{1,2} = 152 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 5 \\ 22 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \rightarrow c_{2,1}$$

untuk baris $x = 2$ dan kolom $y = 2$, dengan entri $p_{2,2} = 116$

$$p_{2,2} = 116 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 6 \\ 26 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \rightarrow c_{0,2}$$

4. Karena iterasi $d = 1$ maka proses transformasi hanya dilakukan satu kali, Sehingga hasil trasformasi menghasilkan matriks C dengan entri sebagai berikut.

$$c_{0,0} = 140, c_{1,0} = 163, c_{2,0} = 146, c_{0,1} = 122, c_{1,1} = 124, c_{2,1} = 152,$$

$$c_{0,2} = 116, c_{1,2} = 95, c_{2,2} = 114$$

Maka diperoleh marik C sebagai berikut.

$$C = \begin{bmatrix} 140 & 122 & 116 \\ 163 & 124 & 95 \\ 146 & 152 & 114 \end{bmatrix}$$

Matriks inilah yang membentuk *cipher-image* hasil dari algoritma *Arnold cat map*.

Gambar 3.4 *Cipher-image grayscale* enkripsi *Arnlod Cat Map*

3.2 Enkripsi Citra dengan Super Enkripsi

3.2.1 Algoritma Enkripsi Citra dengan Super Enkripsi

Pada penelitian ini, super enkripsi akan menggabungkan teknik subsitusi dan tranposisi namun menggunakan algoritma yang berbeda, algoritma *Vinegere cipher* sebagai teknik subsitusi dan *Arnold cat map* sebagai trasposisi.

Supaya mudah memahami dibuatlah ilustrasi sebagai berikut :

1. S memasukkan kunci berupa teks dan tiga bilangan bulat (a, b, d) sebagai *cipher*, serta memasukkan *plain-image* P.
2. Kode ASCII akan menerjemahkan teks kedalam desimal. *Plain-image* P menggunakan bilangan desimal tersebut untuk menjadi *cipher-image-1* dengan algoritma *Vinegere cipher*.
3. *Cipher-image-1* menggunakan nilai a, b dan d akan dibentuk menjadi *cipher-image* dengan algotitma *ACM*.
4. S mengirimkan pesan berupa *cipher-image* kepada R. R akan mendekripsi pesan dengan memasukkan teks yang sama seperti S masukkan dan tiga bilangan bulat (a, b, d).

Secara lebih jelas mengenai algoritma enkripsi dengan menggunakan super enkripsi dijelaskan sebagai berikut :

1. Memasukkan teks sebagai kunci algoritma *Vinegere cipher*, minimal teks terdiri dari 3 karakter dan integer a, b dan d sebagai kunci pada algoritma *Arnold cat map*. Serta menyertakan *plain-image* (citra asli) berukuran $N \times N$ pixel yang akan dienkripsi.
2. Membaca matriks *plain-image* sebagai matriks $N \times N$. *Plain-image*(citra asli) akan dibagi menjadi 3 layer yaitu layer R*(Red)*, *G(green)* dan B *(blue)*. Ketiga layer tersebut akan menciptakan tiga matriks yang berbeda namun memiliki ordo yang sama dengan *plain-image.*
3. Teks akan diterjemahkan kedalam angka dengan kode ASCII, sehingga diperoleh $k_1, k_2,..,k_t$ dengan t=panjang kunci/karakter.
4. Membentuk matriks *plain-image* sebagai array *plain-image*.
5. Membentuk kunci $k_1, k_2,..,k_t$ sebagai array sepanjang array *plain-image.*
6. Setiap kunci $k_1, k_2,..,k_t$ yang berbentuk desimal akan dirubah dalam bentuk *binary*. Begitu pula pada *plain-image,* setiap nilai pixel yang berbentuk desimal dirubah dalam bentuk *binary*.
7. Menjalankan algoritma *Vinegere cipher* dengan mengoperasikan array *plain-image* dan array kunci menggunakan operasi XOR dengan menggunakan persamaan (3.1) sehingga membentuk array dengan entri baru yang dimisalkan sebagai array *cipher*.
8. Array *cipher* akan diubah menjadi matriks *cipher*.
9. Selanjutnya dari matriks *cipher* akan diambil titik koordinat setiap entri, titik koordinat tersebut akan di kalikan dengan matrik 2×2 yang entrinya merupakan *integer* a dan b. Proses transformasi ini menggunakan persamaan (3.2).

10. Transformsi dilakukan terus menerus sampai iterasi *d* dan seluruh pixel dalam terpetakan dan menghasilkan matriks baru yang disebut *cipher-image*. enkripsi dilakukan pada matriks dari tiap layer, R, G dan B maka akan ada tiga matriks yang yang melakukan langkah 3 sampai 10 secara terpisah. Kemudian *cipher-image* akan terbentuk dengan menggabungkan ketiga matriks tersebut secara tertumpuk berturut-turut dari matriks *cipher* dari layer R, G dan B.
11. S mengirimkan *cipher-image* kepada R.

3.2.2 Simulasi Super Enkripsi Citra

3.2.2.1 Simulasi Algoritma Super Enkripsi pada Matriks

Enkripsi dengan metode super enkripsi, akan dilakukan dengan menggabungkan algoritma subsitusi *Vinegere cipher* dengan trasposisi *Arnold cat Map*. Algoritma enkripsi akan berjalan dengan metode subsitusi *Vinegere cipher* terlebih dahulu, berikutnya dilanjutkan dengan metode *Arnold Cat Map*. Dalam enkripsi citra yang menjadi objek kriptografinya adalah *pixel* yang menyusun citra tersebut.

Dalam proses enkripsi dengan *Vinegere cipher* dibutuhkan kunci yang terdiri dari karakter-karakter baik huruf maupun simbol, minimal 3 karakter. Misalkan pilih "Key" sebagai kunci, maka setelah diterjemahkan dalam decimal dengan kode ASCII didapatkan $k_1 = 75$, $k_2 = 101$, $k_3 = 122$, dan *plain-image* berukuran 3×3 sebagai berikut.

Gambar 3.5 *Plain-image grayscale* 3×3 pixel

Gambar 3.5 merupakan citra *grayscale* yang terdiri satu layer saja. Citra diatas jika dibaca sebagai matrik 3×3 maka terbentuk entri sebagai berikut sebagai berikut dengan P adalah matrik *plain-image*.

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

Matrik P akan dirubah kedalam bentuk array satu dimensi,sehingga terbentuk P_1 sebagai berikut.

$$P_1 = \{140,\ 146, 163,\ 124,\ 122,\ 152,\ 114,\ 95,\ 116\}$$

Selanjutnya kunci $k_1 = 75,\ k_2 = 101,\ k_3 = 122$ akan dibentuk sesuai panjang P_1 . Jika panjang kunci tidak sama dengan P_1 maka akan diulang sampai memenuhi panjang P_1

$$K = \{k_1, k_2, k_3, k_1, k_2, k_3, k_1, k_2, k_3\}$$

maka

$$K = \{\ 75,\ 101,\ 122\ , 75,\ 101,\ 122\ , 75,\ 101,\ 122\ \}$$

Kedua himpunan ini akan dimasukkan dalam persamaan *Vinegere cipher* , $p_i \in P_1$ dan $k_j \in K$ dengan $1 \leq i \leq 9$ dan $1 \leq j \leq 3$ sebagai berikut.

$$c_i = p_i \oplus k_j$$

Sehingga enkripsi citra dilakukan sebagai berikut.

$$c_1 = p_1 \oplus k_1 = 140_{10} \oplus 75_{10} = 10001100_2 \oplus 00101011_2 = 10100111_2$$
$$= 167_{10}$$

$$c_2 = p_2 \oplus k_2 = 146_{10} \oplus 101_{10} = 10010010_2 \oplus 01100101_2 = 11110111_2$$
$$= 247_{10}$$

$$c_3 = p_3 \oplus k_3 = 163_{10} \oplus 122_{10} = 10100011_2 \oplus 01111010_2 = 11011001_2$$
$$= 217_{10}$$

$$c_4 = p_4 \oplus k_1 = 124_{10} \oplus 75_{10} = 01111100_2 \oplus 00101011_2 = 01010111_2$$
$$= 87_{10}$$

$$c_5 = p_5 \oplus k_2 = 122_{10} \oplus 101_{10} = 01111010_2 \oplus 01100101_2 = 00011111_2$$
$$= 31_{10}$$

$$c_6 = p_6 \oplus k_3 = 152_{10} \oplus 122_{10} = 10011000_2 \oplus 01111010_2 = 11100010_2$$
$$= 226_{10}$$

$$c_7 = p_7 \oplus k_1 = 114_{10} \oplus 75_{10} = 01110010_2 \oplus 00101011_2 = 01011001_2$$
$$= 89_{10}$$

$$c_8 = p_8 \oplus k_2 = 95_{10} \oplus 101_{10} = 01011111_2 \oplus 01100101_2 = 00111010_2$$
$$= 58_{10}$$

$$c_9 = p_9 \oplus k_3 = 116_{10} \oplus 122_{10} = 01110100_2 \oplus 01111010_2 = 00001110_2$$
$$= 28_{10}$$

Dari hasil subsitusi *Vinegere cipher* didapatkan $c_i \in C_1$ dengan $1 \leq i \leq 9$ sehingga terbentuk array C sebagai berikut.

$$C_1 = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$$

Maka $C_1 = \{167,247,217,87,31,226,89,58,28\}$

Dari C_1diatas dapat dibentuk Matriks C sebagai berikut :

$$C = \begin{bmatrix} 167 & 247 & 217 \\ 87 & 31 & 226 \\ 89 & 58 & 28 \end{bmatrix}$$

Untuk memperkuat algoritmanya, Selanjutnya matriks tersebut akan diacak entrinya menggunakan algoritma dari *Arnold cat map*. Jika $c_{x,y} \in C$, dengan $x = 0,1,2$ dan $y = 0,1,2$ .Misalkan $a = 2$ dan $b = 4$ dengan iterasi $d = 1$, diperoleh sebagai berikut.

Berikut proses transformasi titik matrik C pada matrik H. jika $c_{x,y} \epsilon C$ dan $h_{x,y} \in H$ dengan baris $x = 0,1,2$ dan kolom $y = 0,1,2$

untuk baris $x = 0$ dan kolom $y = 0$, dengan entri $c_{0,0} = 167$

$$c_{0,0} = 167 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow h_{0,0}$$

untuk baris $x = 1$ dan kolom $y = 0$, dengan entri $c_{1,0} = 87$

$$c_{1,0} = 87 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 4 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow h_{1,1}$$

untuk baris $x = 2$ dan kolom $y = 0$, dengan entri $c_{2,0} = 89$

$$c_{2,0} = 89 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 8 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow h_{2,2}$$

untuk baris $x = 0$ dan kolom $y = 1$, dengan entri $c_{0,1} = 247$

$$c_{0,1} = 247 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 9 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \rightarrow h_{0,2}$$

untuk baris $x = 1$ dan kolom $y = 1$, dengan entri $c_{1,1} = 31$

$$c_{1,1} = 31 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 3 \\ 13 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow h_{0,1}$$

untuk baris $x = 2$ dan kolom $y = 1$, dengan entri $c_{2,1} = 58$

$$c_{2,1} = 58 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 4 \\ 17 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow h_{1,2}$$

untuk baris $x = 0$ dan kolom $y = 2$, dengan entri $c_{0,2} = 217$

$$c_{0,2} = 217 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 4 \\ 9 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow h_{1,0}$$

untuk baris $x = 1$ dan kolom $y = 2$, dengan entri $c_{0,0} = 247$

$$c_{1,2} = 226 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 5 \\ 22 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \rightarrow h_{2,1}$$

untuk baris $x = 2$ dan kolom $y = 2$, dengan entri $c_{2,2} = 28$

$$c_{2,2} = 28 \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 6 \\ 26 \end{bmatrix} \ (mod\ 3) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \rightarrow h_{0,2}$$

Hasil dari perkalian matriks $\begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}$ dengan entri pada matrik $C = [c_{nn}]$ dengan $0 \leq n \leq 2$ akan menghasilkan matriks $H = [h_{nn}]$. Sehingga *matrik H* dapat dibentuk sebagai berikut.

$$H = \begin{bmatrix} 167 & 31 & 28 \\ 217 & 87 & 58 \\ 247 & 226 & 89 \end{bmatrix}$$

Matrik *H* ini lah yang akan membentuk *cipher-image* berikut ini .

Gambar 3.6 *Cipher-image Grayscale* enkripsi Super Enkripsi

3.2.2.2 Simulasi Enkripsi Citra dengan Super Enkripsi pada GUI MATLAB

Dalam merancang program enkripsi dengan GUI MATLAB diperlukan suatu gambaran jalannya suatu program, gambaran tersebut dapat berupa algoritma maupun *flowchart* yang akan digunakan sebagai dasar rancangan kriptografi. Algoritma merupakan suatu tahapan-tahapan yang dilakukan secara

urut dan jelas. Berikut ini pembentukan algoritma super enkripsi dalam program MATLAB .

INPUT : *Plain-image* $N \times N$(citra asli), password(kunci text), tiga integer

OUTPUT: *Cipher-image* $N \times N$

PROSES:

STEP I : Mengubah/membaca *Plain-image* $N \times N$ sebagai Matriks $N \times N$.
Plain-image $N \times N \rightarrow layer\ Red$, $Green$ dan *Blue*
Layer red $\rightarrow Matriks\ R_{N\times N}$
$layer\ Green \rightarrow Matriks\ G_{N\times N}$
Layer Blue $\rightarrow Matriks\ B_{N\times N}$

STEP II : Mengubah Matriks menjadi array dimensi satu/vektor.
$Matriks\ R_{N\times N} \rightarrow\ Vektor\ R_{1\times N^2}$
$Matriks\ G_{N\times N} \rightarrow Vektor\ G_{1\times N^2}$
$Matriks\ B_{N\times N} \rightarrow Vektor\ B_{1\times N^2}$

STEP III: Mengubah *password* kedalam bilangan integer positif dengan kode ASCII
Password $\rightarrow k_1, k_2, \dots, k_t$
$t = panjang\ karakter\ password$
$k \in K\ , k\ \in Z^+$

STEP IV : Menyamakan panjang K dengan vektor *plain-image* $R/G/B$.
$Vektor\ R = r_1, r_2, r_3, r_4\ \dots, r_{N^2}$
$Vektor\ G = g_1, g_2, g_3, g_4\ \dots, g_{N^2}$
$Vektor\ B = b, b_2, b_3, b_4\ \dots, b_{N^2}$
$K = k_1, k_2, .., k_t, k_1, k_2, \dots, k_{N^2}$

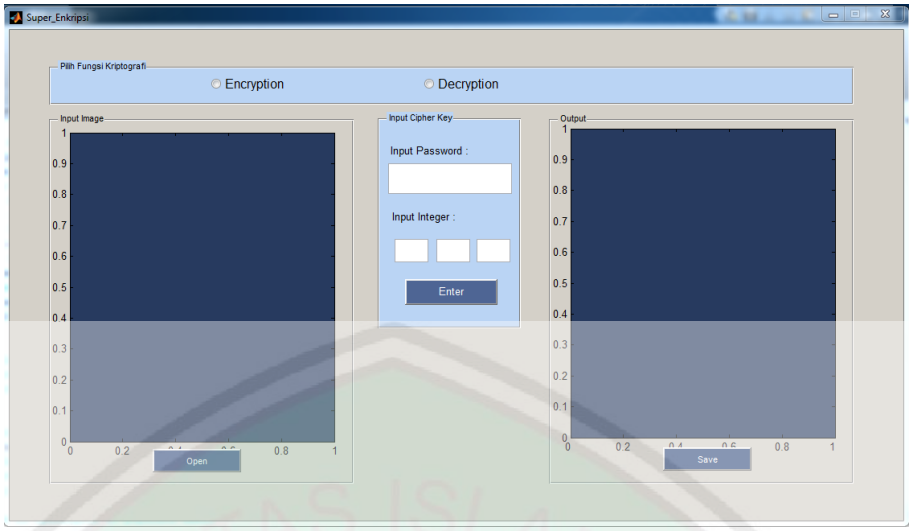
STEP V : Mengoperasikan $Vektor\ R, G\ dan\ B$ dengan K menggunakan operasi XOR sehingga diperoleh vektor $R_1, G_1\ dan\ B_1$

STEP VI : Mengubah vektor $R_1, G_1\ dan\ B_1$ menjadi matriks dengan ordo yang sesuai *plain-image*, sehingga diperoleh matriks $R_2, G_2\ dan\ B_2$.

STEP VII : Melakukan transposisi pada matriks $R_2, G_2\ dan\ B_2$ dengan *Arnold Cat Map*, sehingga diperoleh matriks $R_3, G_3\ dan\ B_3$ dengan menggunakan tiga integer yang telah diinput.

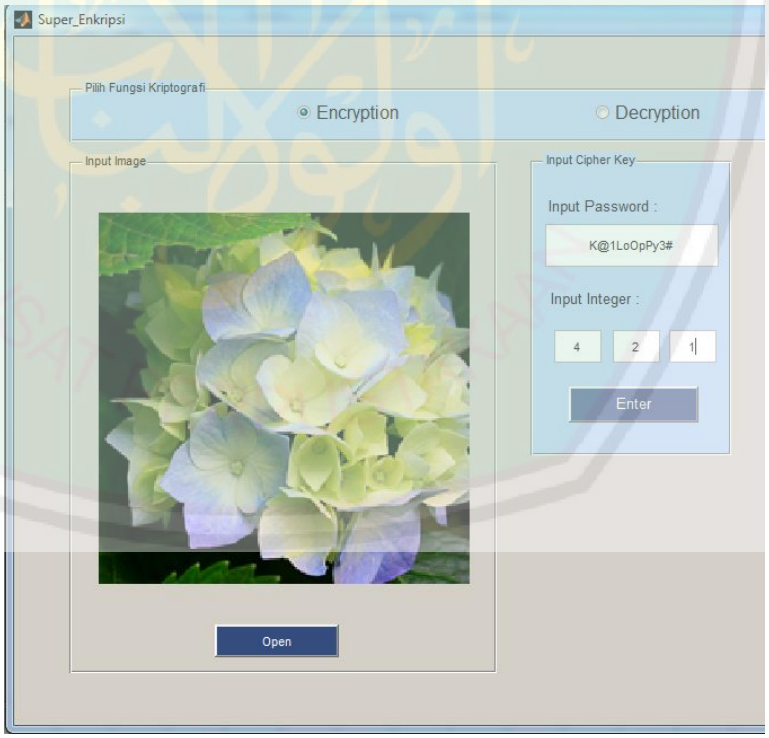
STEP VIII: Menggabungkan matriks $R_3, G_3\ dan\ B_3$ sehingga membentuk *cipher-image.*

Berikut ini adalah simulasi dengan menggunakan aplikasi MATLAB. Dengan melakukan simulasi dengan aplikasi dapat dilihat perbedaan antara *plain-image* dengan *cipher-image.*



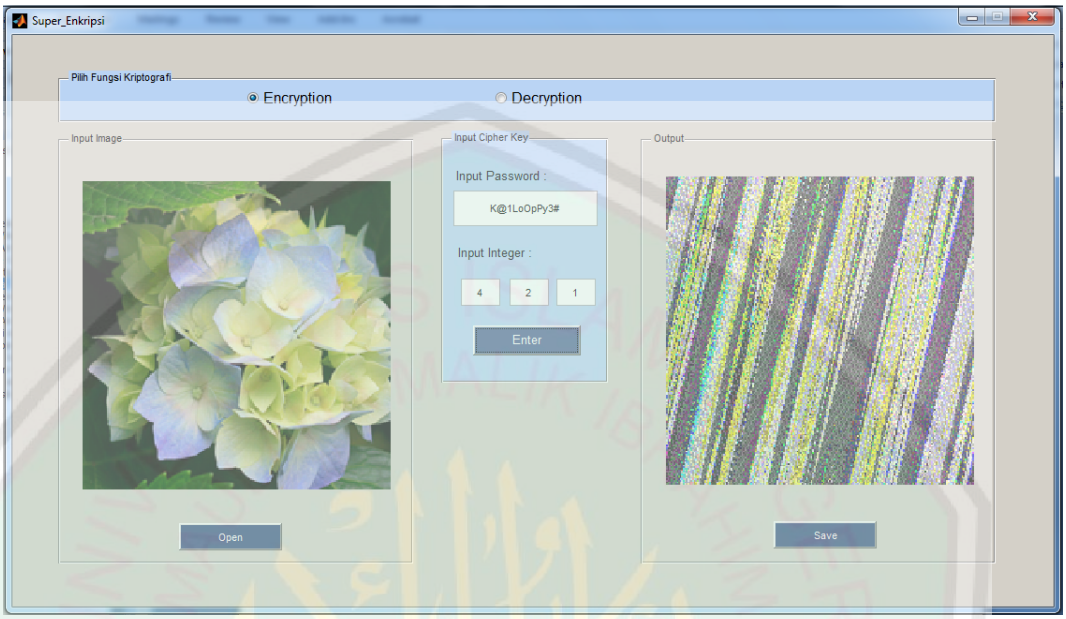
Gambar 3.7 *interface* aplikasi super enkripsi

Pada gambar 3.7 untuk melakukan enkripsi, langkah awal pengguna memilih fungsi baik enkripsi maupun dekripsi sesuai tujuan. Selanjutnya memasukkan kunci berupa password dan 3 integer serta *plain-image* seperti gambar 3.8 berikut.



Gambar 3.8 enkripsi citra dengan MATLAB

Setelah memasukkan *plain-image* dan *cipher-key* berupa password dan integer, langkah selanjutnya menekan enter untuk melakukan enkripsi pada *plain-image*. Sehinga diperoleh hasil pada gambar 3.9.



Gambar 3.9 Hasil enkripsi

Hasil citra pada *output* inilah yang disebut dengan *cipher-image*. *Cipher-image* tersebut dibentuk dengan menggunakan *password* K@1LoOpPy3# dan *integer* $a = 4, b = 2$ dan iterasi sebanyak 1.

3.3 Analisa dekripsi *Vinegere cipher* dan *Arnold Cat Map*

3.3.1 Algoritma Dekripsi dengan Algoritma *Vinegere Cipher*

Langkah penting yang harus dilakukan setelah melakukan enkripsi adalah dekripsi. Melalui dekripsi ini hasil enkripsi akan diubah kembali kebentuk aslinya, maka dalam penelitian ini *cipher-image* akan dirubah ke dalam *plain-image* dengan proses dekripsi.

Misalkan P merupakan matriks dari *plain-image* , C matriks *cipher-image*. Misalkan matriks P dan C memiliki ordo $N \times N$, Matriks-matriks

tersebut kemudian disusun menjadi array dimensi 1 dengan mengambil baris ke-1 sampai ke-N. Jika $q \in P$ dan $r \in C$ maka array dari P dan C dapat disusun sebagai berikut.

$$\{q_{1,1}, q_{1,2}, \dots, q_{1,N}, q_{2,1}, q_{2,2}, \dots, q_{2,N}, \dots, q_{N,1}, q_{N,2}, \dots, q_{N,N}\}$$

$$\underbrace{\qquad}_{q_{1,n}} \quad \underbrace{\qquad}_{q_{2,n}} \quad \dots \quad \underbrace{\qquad}_{q_{N,n}}$$

Dengan $n = 1,2, \dots, N$. Begitu pula dengan penyusunan array pada matriks C

$$\{r_{1,1}, r_{1,2}, \dots, r_{1,N}, r_{2,1}, r_{2,2}, \dots, r_{2,N}, \dots, r_{N,1}, r_{N,2}, \dots, r_{N,N}\}$$

Maka jika dituliskan kembali entri di dalam P_1 dan C_1 yang berturut-turut merupakan array dimensi satu dari P dan C akan memiliki panjang N^2 sebagai berikut.

$$P_1 = \{p_1, p_2, p_3, \dots, p_{N^2}\}$$

$$C_1 = \{c_1, c_2, c_3, \dots, c_{N^2}\}$$

Jika $p_i \in P_1$, $c_i \in P_1$ dan $k_j \in K$ yang merupakan himpunan kunci dengan $1 \leq i \leq N^2$ serta $1 \leq j \leq t$ dengan $t =$ panjang kunci. Maka diproleh dekripsi dengan algoritma *Vinegere cipher* modifikasi *XOR* sebagai berikut.

$$p_i = c_i \oplus k_j \tag{3.3}$$

Keterangan :

$p = nilai\ pixel\ pada\ plain - image$

$c = nilai\ pixel\ pada\ cipher\ image$

$k = nilai\ kunci\ dalam\ desimal$

$\oplus = notasi\ operasi\ Exclusive - OR$

$i = 1,2,,3, \dots, N^2$

$j = 1,2,3, \dots, t$

$N = ordo\ kolom/baris\ citra$

$t = panjang\ kunci$

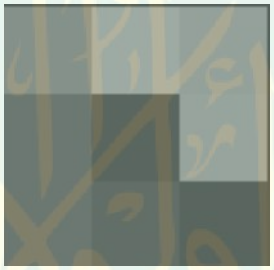
Berikut ini algoritma dalam melakukan dekripsi dengan menggunakan algoritma *Vinegere cipher* modifikasi *XOR.*

1. Memasukkan teks sebagai kunci algoritma *Vinegere cipher,* yang telah digunakan pada proses enkripsi. Serta menyertakan *cipher-image* (citra asli) berukuuran $N \times N$ pixel yang telah dienkrispi.
9. Membaca matriks *cipher-image* sebagai matriks $N \times N$. *Cipher-image*(citra asli) dibagi menjadi 3 layer yaitu layer R*(Red)*, *G(green)* dan B *(blue).* Setiap layer akan membetuk matriks yang berbeda.
2. Membentuk matriks *plain-image* sebagai array *plain-image*.
3. Teks akan diterjemahkan kedalam angka dengan kode ASCII, jika $t =$ panjang kunci maka diperoleh $k_1, k_2, \dots, k_t$.
4. Membentuk kunci $k_1, k_2, \dots, k_t$ sebagai array sepanjang array *plain-image.*
5. Setiap kunci $k_1, k_2, \dots, k_t$ yang berbentuk decimal akan dirubah dalam bentuk *binary*. Begiru pula pada *cipher-image,* setiap nilai pixel yang berbentuk desimal dirubah dalam bentuk *binary*.
6. Menjalankan algoritma *Vinegere cipher* dengan mengoperasikan matriks *plain-image* dan matriks kunci menggunakan operasi XOR dengan menggunakan persamaan (3.3) sehingga membentuk array dengan entri baru yang dimisalkan sebagai array *plain*.
10. Array *plain* akan diubah menjadi matriks *plain*. Matriks *plain* ini kemudian akan membentuk citra *plain-image*. Dekripsi dilakukan pada tiga matriks dengan melakukan langkah 3 sampai 8 secara terpisah. Kemudian *plain-*

image akan terbentuk dengan menggabungkan ketiga matriks tersebut secara tertumpuk berturut-turut dari matriks *plain* dari layer R, G dan B.

Berikut ini penerapan dekripsi algoritma *Vinegere cipher* modifikasi XOR pada *plain-image* berukuran 3×3 pixel. Penerapan dilakukan pada *cipher-image grayscale*, sebuah citra *grayscale* dapat tersusun dengan satu layer saja. Walaupun pada umumnya semua citra digital baik *grayscale* atau berwarna tersusun atas layer RGB. Pada contoh berikut ini digunakan citra *grayscale* tersusun satu layer yang telah dienkripsi pada subbab 3.1.1.

1. Misalkan *cipher-image* berukuran 3×3 pixel akan di enkripsi dengan kunci "Key" sebagai berikut.



Gambar 3.10 *Cipher-image Grayscale* enkripsi *Vinegere Cipher*

2. Selanjutnya cipher-*image* berukuran 3×3 pixel dibaca sebagai matrik berordo 3×3 sebagi berikut.

$$C = \begin{bmatrix} 167 & 247 & 217 \\ 87 & 31 & 226 \\ 89 & 58 & 28 \end{bmatrix}$$

3. Kunci "Key" diterjemahkan dalam desimal dengan bantuan kode ASCII didapatkan $k_1 = 75,\ k_2 = 101,\ k_3 = 122.$
4. Membentuk array dari matriks *C* sehingga diperoleh

   $C_1 = \{167,\ 247, 217,\ 87,\ 31,\ 226,\ 89,\ 58,\ 28\}$
5. Membentuk array dengan entri $k_1, k_2, dan\ k_3$ dengan mengulang entri tersebut sepanjang C_1 . Maka

$$K = \{k_1, k_2, k_3, k_1, k_2, k_3, k_1, k_2, k_3\}$$

Sehingga

$$K = \{\ 75, 101, 122, 75, 101, 122, 75, 101, 122\ \}$$

6. Jika setiap anggota dari himpunan C_1 dan K dirubah kedalam bentu angka binary dengan $c_i \in C_1$ dan $k_j \in K$ dengan $1 \leq i \leq 9, 1 \leq j \leq 3$ maka diperoleh

$$c_1 = 167_{10} = 10100111_2$$

$$c_2 = 247_{10} = 11110111_2$$

$$c_3 = 217_{10} = 11011001_2$$

$$c_4 = 87_{10} = 01010111_2$$

$$c_5 = 31_{10} = 00011111_2$$

$$c_6 = 226_{10} = 11100010_2$$

$$c_7 = 89_{10} = 01011001_2$$

$$c_8 = 58_{10} = 00111010_2$$

$$c_9 = 28_{10} = 00001110_2$$

Serta nilai entri K sebagai berikut.

$$k_1 = 75_{10} = 00101011_2$$

$$k_2 = 101_{10} = 01100101_2$$

$$k_3 = 122_{10} = 01111010_2$$

7. Dengan menggunakan persamaan (3.3) diperoleh hasil berikut.

$$p_1 = c_1 \oplus k_1 = 167_{10} \oplus 75_{10} = 10100111_2 \oplus 00101011_2$$

$$= 10001100_2 = 140_{10}$$

$$p_2 = c_2 \oplus k_2 = 247_{10} \oplus 101_{10} = 11110111_2 \oplus 01100101_2 = 10010010_2$$

$$= 146_{10}$$

$$p_3 = c_3 \oplus k_3 = 217_{10} \oplus 122_{10} = 11011001_2 \oplus 01111010_2 = 10100011_2$$
$$= 163_{10}$$

$$p_4 = c_4 \oplus k_1 = 87_{10} \oplus 75_{10} = 01010111_2 \oplus 00101011_2 = 01111100_2$$
$$= 124_{10}$$

$$p_5 = c_5 \oplus k_2 = 31_{10} \oplus 101_{10} = 00011111_2 \oplus 01100101_2 = 01111010_2$$
$$= 122_{10}$$

$$p_6 = c_6 \oplus k_3 = 226_{10} \oplus 122_{10} = 11100010_2 \oplus 01111010_2 = 10011000_2$$
$$= 152_{10}$$

$$p_7 = c_7 \oplus k_1 = 89_{10} \oplus 75_{10} = 01011001_2 \oplus 00101011_2 = 01110010_2$$
$$= 114_{10}$$

$$p_8 = c_8 \oplus k_2 = 58_{10} \oplus 101_{10} = 00111010_2 \oplus 01100101_2 = 01011111_2$$
$$= 95_{10}$$

$$p_9 = c_9 \oplus k_3 = 28_{10} \oplus 122_{10} = 00001110_2 \oplus 01111010_2 = 01110100_2$$
$$= 116_{10}$$

Dari hasil subsitusi diatas diperoleh P_1

$$P_1 = \{140,146,163,124,122,152,114,95,116\}$$

8. P_1 akan dirubah kedalam bentuk matriks kembali dengan ordo yang sama dengan matriks C.

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

Matrik P inilah yang akan menjadi dasar pembentukan citra asli yang disebut dengan *plain-image.*

Gambar 3.11 . *Plain-image grayscale*. Dekripsi *Vinegere Cipher*

3.3.2 Algoritma Dekripsi dengan *Arnold Cat Map*

Pada algoritma enkripsi *Arnold Cat Map* melakukan transformasikan titik pada *plain-image* (x, y) pada titik lain (x', y'). Maka pada dekripsinya titik (x', y') akan ditranformasikan pada titik aslinya (x, y). sehingga diperoleh persamaan dekripsi sebagai berikut.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} (mod\ N) \tag{3.4}$$

Keterangan :

$x = titik\ awal\ pixel\ di\ koordinat\ x$

$y = titik\ awal\ \ pixel\ di\ koordinat\ y$

$x' = titik\ pixel\ hasil\ \ tranformasi\ di\ koordinat\ x$

$y' = titik\ pixel\ hasil\ transformasi\ di\ koordinat\ y$

$a, b = bilangan\ bulat\ positif$

$N = \ ordo\ matriks\ citra\ dalam\ kolom\ \ / baris$

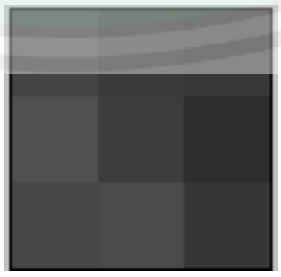
Berikut ini algoritma dalam melakukan dekripsi dengan menggunakan algoritma *Arnold cat map.*

1. Memasukkan kunci berupa tiga integer (a, b, d) yang telah digunakan dalam enkripsi. Serta menyertakan *citra-image* (citra terenkripsi) berukuran $N \times N$ pixel yang telah dienkrispi.
2. Membaca *cipher-image* berukuran sebagai matriks $N \times N$

3. Masukkan integer a dan b dalam persamaan (3.4).
4. Selanjutnya dari matriks $N \times N$ akan diambil titik koordinat setiap entri, titik koordinat tersebut akan di kalikan dengan matrik 2×2 yang entrinya merupakan *integer* a dan b. Proses transformasi ini menggunakan persamaan (3.4).
5. Transformsi dilakukan terus menerus sampai iterasi d dan seluruh pixel dalam terpetakan ketitiknya asal dan menghasilkan matriks asal sebelum di enkripsi yang disebut *plain-image*.

Berikut ini penerapan algoritma *Arnold cat Map* pada *cipher-image* berukuran 3×3 pixel. Penerapan dilakukan pada *cipher-image grayscale*, sebuah citra *grayscale* dapat tersusun dengan satu layer saja. Walaupun pada umumnya semua citra digital baik *grayscale* atau berwarna tersusun atas layer RGB. Pada contoh berikut ini digunakan citra *grayscale* tersusun satu layer yang telah dienkripsi pada subbab 3.1.2.

1. Kunci yang digunakan dalam dekripsi harus sama dengan yang digunakan dalam enkripsi. Misalkan kunci yang digunakan dalam enkripsi citra sebelumnya adalah $= 2$, $b = 4$ dan $d = 1$. Dengan *cipher-image* berukuran 3×3 sebagai berikut.



Gambar 3.12 *Cipher-image grayscale* enkripsi *Arnold Cat Map*

2. Selanjutnya *cipher-image* berukuran 3×3 akan dibaca sebagai matriks 3×3 sebagai berikut.

$$C = \begin{bmatrix} 140 & 122 & 116 \\ 163 & 124 & 95 \\ 146 & 152 & 114 \end{bmatrix}$$

3. Memasukkan kunci ke dalam persamaan (3.4).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} (mod\ N)$$

$$= \frac{1}{9 \cdot 1 - 4 \cdot 2} \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} (mod\ 3)$$

$$= \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} (mod\ 3)$$

4. Berikut proses transformasi titik matrik C pada matrik P. jika $c_{x,y} \epsilon C$ dan $p_{x,y} \in P$ dengan baris $x = 0,1,2$ dan kolom $y = 0,1,2$

untuk baris $x = 0$ dan kolom $y = 0$, dengan entri $c_{0,0} = 140$

$$c_{0,0} = 140 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow p_{0,0}$$

untuk baris $x = 1$ dan kolom $y = 0$, dengan entri $c_{1,0} = 163$

$$c_{1,0} = 163 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \rightarrow p_{0,2}$$

untuk baris $x = 2$ dan kolom $y = 0$, dengan entri $c_{2,0} = 146$

$$c_{2,0} = 146 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow p_{0,1}$$

untuk baris $x = 0$ dan kolom $y = 1$, dengan entri $c_{0,1} = 122$

$$c_{0,1} = 122 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow p_{1,1}$$

untuk baris $x = 1$ dan kolom $y = 1$, dengan entri $c_{1,1} = 124$

$$c_{1,1} = 124 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow p_{1,0}$$

untuk baris $x = 2$ dan kolom $y = 1$, dengan entri $c_{2,1} = 152$

$$c_{2,1} = 152 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow p_{1,2}$$

untuk baris $x = 0$ dan kolom $y = 2$, dengan entri $c_{0,2} = 116$

$$c_{0,2} = 116 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow p_{2,2}$$

untuk baris $x = 1$ dan kolom $y = 2$, dengan entri $c_{1,2} = 95$

$$c_{1,2} = 95 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \rightarrow p_{2,1}$$

untuk baris $x = 2$ dan kolom $y = 2$, dengan entri $c_{2,2} = 114$

$$c_{2,2} = 114 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \rightarrow p_{2,0}$$

5. Karena iterasi $d = 1$ maka proses transformasi hanya dilakukan satu kali, Sehingga hasil trasformasi entri yang akan membentuk matrik P, dengan entri sebagai berikut.

$$p_{0,0} = 140, p_{1,0} = 124, p_{2,0} = 114, p_{0,1} = 146, p_{1,1} = 122, p_{2,1} = 95,$$

$$p_{0,2} = 163, p_{1,2} = 95, p_{2,2} = 116$$

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

Matriks P inilah yang membentuk *plain-image* dengan entri-entri dalam matriks P yang membentuk nilai pixel dari *plain-image* (citra asli).

Gambar 3.13 *plain-image grayscale* dekripsi *Arnold Cat Map*

3.4 Dekripsi Citra dengan Super Enkripsi

3.4.1 Algoritma Dekripsi dengan Super Enkripsi

Pada Algoritma enkripsi 3.2.1 super enkripsi dilakukan dengan menjalankan *Vinegere cipher* terlebih dahulu selanjutnya dilanjutkan dengan Arnold cat map. Namun pada algoritma dekripsinya akan menggunakan sebaliknya. Dekripsi bertujuan mengubah *cipher-image* menjadi *plain-image.* Secara lebih jelas mengenai dekripsi super enkripsi dengan *Arnold cat map* dan *Vinegere cipher* dijelaskan sebagai berikut :

1. Memasukkan teks sebagai kunci algoritma *Vinegere cipher*, minimal teks terdiri dari 3 karakter dan integer a, b dan d sebagai kunci pada algoritma *Arnold cat map*. Serta menyertakan *cipher-image* berukuuran $N \times N$ pixel yang akan didekripsi. Kunci merupkan kunci yang digunakan dalam proses enkripsi *cipher-image* sebelumnya.
2. Membaca *cipher-image* berukuran sebagai matriks $N \times N$. *cipher-image*(citra asli) memiliki tiga layer yaitu layer R*(Red), G(green)* dan B *(blue).*
3. Masukkan integer b dan c dalam persaman (3.4).
4. Selanjutnya dari matriks $N \times N$ akan diambil titik koordinat setiap entri, titik koordinat tersebut akan di kalikan dengan matrik 2×2 yang entrinya merupakan *integer* a dan b. Proses transformasi ini menggunakan persamaan (3.4).
5. Transformsi dilakukan terus menerus sampai iterasi *d* dan seluruh pixel dalam terpetakan ketitiknya asal dan menghasilkan matriks baru dan menghasilkan *plain-image-1*. Selanjutnya pada tahap ke-6, super enkripsi akan dilakukan oleh algoritma *Vinegere cipher.*

6. Teks akan diterjemahkan kedalam angka dengan kode ASCII, jika kunci yang digunakan hanya berupa tiga alphabet maka diperoleh k_1, k_2 dan k_3.
7. Membentuk kunci k_1, k_2 dan k_3 sebagai array sepanjang array *plain-image-1*.
8. Setiap kunci k_1, k_2 dan k_3 yang berbentuk decimal akan dirubah dalam bentuk *binary*. Begiru pula pada *plain-image-1*, setiap nilai pixel yang berbentuk desimal diubah dalam bentuk *binary*.
9. Menjalankan algoritma *Vinegere cipher* dengan mengoperasikan matriks *plain-image* dan matriks kunci menggunakan operasi XOR dengan menggunakan persamaan (3.3) sehingga membentuk array plain dengan entri baru.
10. Array *plain* akan diubah menjadi matriks *plain*. Matriks *plain* ini kemudian akan membentuk citra *plain-image*. dekripsi dilakukan dengan menggunakan citra RGB maka akan ada tiga matriks yang yang melakukan langkah 3 sampai 10 secara terpisah. Kemudian *plain-image* akan terbentuk dengan menggabungkan ketiga matriks tersebut secara tertumpuk berturut-turut dari matriks *plain* dari layer R, G dan B.

3.4.2 Simulasi Dekripsi Citra dengan Super enkripsi

3.4.2.1 Simulasi Dekripsi dengan Super enkripsi pada Matriks

Enkripsi ini menggunakan metode kunci simetri, maka kunci yang digunakan untuk dekripsi sama dengan kunci yang digunakan dalam enkripsi. Masukkan $a = 2$ dan $b = 4$ dengan iterasi $= 1$, masukkan integer ini dalam persamaan (3.4) dan misalkan *cipher-image* memiliki nilai sebagai berikut.

Gambar 3.14 . *cipher-image grayscale* enkripsi Super enkripsi

Jika Gambar 3.14 merupakan *cipher-image grayscale* yang tersusun atas satu layer saja. Jika dibaca dalam bentuk matriks maka warna-warna tersebut akan memiliki entri sebagai berikut, dengan H adalah matriks dari cipher image *grayscale* 3.15.

$$H = \begin{bmatrix} 167 & 31 & 28 \\ 217 & 87 & 58 \\ 247 & 226 & 89 \end{bmatrix}$$

Maka dengan *ACM* diperoleh perhitungan sebagai berikut :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} \ mod\ 3$$

$$= \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} mod\ 3$$

$$= \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} mod\ 3$$

Selanjutnya proses transformasi titik matrik H pada matrik C. jika $h_{x,y} \epsilon H$ dan $c_{x,y} \in C$ dengan baris $x = 0,1,2$ dan kolom $y = 0,1,2$

untuk baris $x = 0$ dan kolom $y = 0$, dengan entri $h_{0,0} = 167$

$$h_{0,0} = 167 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow c_{0,0}$$

untuk baris $x = 1$ dan kolom $y = 1$, dengan entri $h_{1,1} = 87$

$$h_{1,1} = 87 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow c_{1,0}$$

untuk baris $x = 2$ dan kolom $y = 2$, dengan entri $h_{2,2} = 89$

$$h_{2,2} = 89 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \rightarrow c_{2,0}$$

untuk baris $x = 2$ dan kolom $y = 0$, dengan entri $h_{2,0} = 247$

$$h_{2,0} = 247 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow c_{0,1}$$

untuk baris $x = 0$ dan kolom $y = 1$, dengan entri $h_{0,1} = 31$

$$h_{0,1} = 31 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow c_{1,1}$$

untuk baris $x = 1$ dan kolom $y = 2$, dengan entri $h_{1,2} = 58$

$$h_{1,2} = 58 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \rightarrow c_{2,1}$$

untuk baris $x = 1$ dan kolom $y = 0$, dengan entri $h_{1,0} = 217$

$$h_{1,0} = 217 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \rightarrow c_{0,2}$$

untuk baris $x = 2$ dan kolom $y = 1$, dengan entri $h_{2,1} = 226$

$$h_{2,1} = 226 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow c_{1,2}$$

untuk baris $x = 0$ dan kolom $y = 2$, dengan entri $h_{0,2} = 28$

$$h_{0,2} = 28 \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 9 & -2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} (mod\ 3) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow c_{2,2}$$

Dari hasil perkalian matriks dengan (x', y') diperoleh c_{nn} dengan $0 \leq n \leq 2$ dan $c_{nn} \in C$ sebagai berikut :

$$C = \begin{bmatrix} 167 & 247 & 217 \\ 87 & 31 & 226 \\ 89 & 58 & 28 \end{bmatrix}$$

Selanjutnya matriks C tersebut akan didekripsi dengan algoritma *Vinegere Cipher* dengan memasukkan "Key" sebagai kunci dekripsi. matriks C

akan disusun menjadi array satu dimensi terlebih dahulu sehingga terbentuk C_1 sebagai berikut.

$$C_1 = [167, 247, 217, 87, 31, 226, 89, 58, 28\}$$

Kunci "Key" diterjemahkan dalam desimal dengan kode ASCII didapatkan $k_1 = 75$, $k_2 = 101$, $k_3 = 122$. Selanjutnya kunci akan dibentuk sesuai panjang C_1. Jika panjang kunci tidak sama dengan C_1 maka akan diulang sampai memenuhi panjang C_1.

$$K = \{\ 75,\ 101,\ 122,\ 75,\ 101,\ 122,\ 75,\ 101,\ 122\ \}$$

Kedua himpunan ini akan dimasukkan dalam dekripsi *Vinegere cipher* , dengan persamaan dengan modifikasi XOR sebagai berikut.

$$p_i = c_i \oplus k_i$$

Sehingga enkripsi citra dilakukan sebagai berikut.

$$p_1 = c_1 \oplus k_1 = 167_{10} \oplus 75_{10} = 10100111_2 \oplus 00101011_2 = 10001100_2 = 140_{10}$$

$$p_2 = c_2 \oplus k_2 = 247_{10} \oplus 101_{10} = 11110111_2 \oplus 01100101_2 = 10010010_2 = 146_{10}$$

$$p_3 = c_3 \oplus k_3 = 217_{10} \oplus 122_{10} = 11011001_2 \oplus 01111010_2 = 10100011_2 = 163_{10}$$

$$p_4 = c_4 \oplus k_1 = 87_{10} \oplus 75_{10} = 01010111_2 \oplus 00101011_2 = 01111100_2 = 124_{10}$$

$$p_5 = c_5 \oplus k_2 = 31_{10} \oplus 101_{10} = 00011111_2 \oplus 01100101_2 = 01111010_2 = 122_{10}$$

$$p_6 = c_6 \oplus k_3 = 226_{10} \oplus 122_{10} = 11100010_2 \oplus 01111010_2 = 10011000_2 = 152_{10}$$

$$p_7 = c_7 \oplus k_1 = 89_{10} \oplus 75_{10} = 01011001_2 \oplus 00101011_2 = 01110010_2$$
$$= 114_{10}$$

$$p_8 = c_8 \oplus k_2 = 58_{10} \oplus 101_{10} = 00111010_2 \oplus 01100101_2 = 01011111_2$$
$$= 95_{10}$$

$$p_9 = c_9 \oplus k_3 = 28_{10} \oplus 122_{10} = 00001110_2 \oplus 01111010_2 = 01110100_2$$
$$= 116_{10}$$

Hasil dari subsitusi dengan *Vinegere cipher* diperoleh P_1, sebagai berikut.

$$P_1 = \{140, 146, 163, 124, 122, 152, 114, 95, 116\}$$

P_1 diatas akan dibentuk menjadi matrik lagi sesuai ordo matrik C, sehingga terbentuk matrik P sebagai berikut.

$$P = \begin{bmatrix} 140 & 146 & 163 \\ 124 & 122 & 152 \\ 114 & 95 & 116 \end{bmatrix}$$

Jika matriks ini diubah ke dalam bentuk citra maka gambar yang terbuat adalah sebagai berikut.

Gambar 3.15 *Plain-image Grayscale* Dekripsi Super Enkripsi

3.4.2.2 Simulasi Dekripsi dengan Super Enkripsi pada GUI MATLAB

Dalam subbab 3.2.2.2 telah dirancang algoritma enkripsi dengan super enkripsi pada MATLAB. Jika enkripsi merupakan tahap mengubah *plain-image*

menjadi *cipher-image*, maka dekripsi merupakan kebalikannya . Berikut ini pembentukan algoritma dekripsi super enkripsi dalam program MATLAB .

INPUT : *Cipher -image* $N \times N$(citra asli), password(kunci text), tiga integer

OUTPUT: *Plain-image* $N \times N$

PROSES:

STEP I : Mengubah/membaca *Plain-image* $N \times N$ sebagai Matriks $N \times N$.
cipher-image $N \times N \rightarrow layer\ Red$, $Green$ dan *Blue*
$Layer\ red \rightarrow Matriks\ R3_{N\times N}$
$layer\ Green \rightarrow Matriks\ G3_{N\times N}$
$Layer\ Blue \rightarrow Matriks\ B3_{N\times N}$

STEP II : melakukan transposisi pada matriks $R, G, dan\ B$ dengan *Arnold Cat Map*, sehingga diperoleh matriks $R3, G3\ dan\ B3$ dengan menggunakan tiga integer yang telah diinput.

STEP III : Mengubah Matriks menjadi array dimensi satu/vektor.
$Matriks\ R3_{N\times N} \rightarrow\ Vektor\ R3_{1\times N^2}$
$Matriks\ G3_{N\times N} \rightarrow Vektor\ G3_{1\times N^2}$
$Matriks\ B3_{N\times N} \rightarrow Vektor\ B3_{1\times N^2}$

STEP IV: Mengubah *password* kedalam bilangan integer positif dengan kode ASCII
Password $\rightarrow k_1, k_2, \dots, k_t$
$t = panjang\ karakter\ password$
$k \in K\ , k\ \in Z^+$

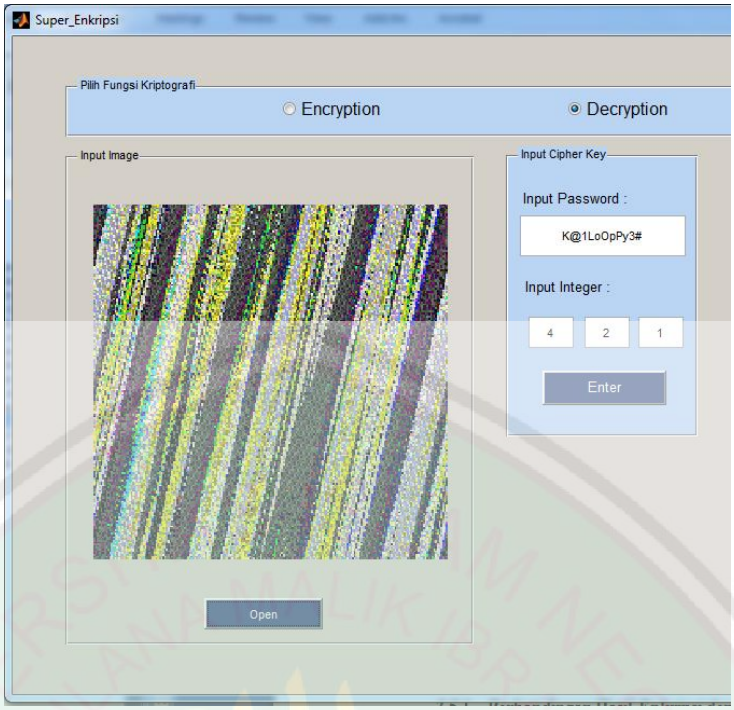
STEP V : Menyamakan panjang K dengan vektor *cipher-image*
$Vektor\ R3 = r_1, r_2, r_3, r_4\ \dots, r_{N^2}$
$Vektor\ G3 = g_1, g_2, g_3, g_4\ \dots, g_{N^2}$
$Vektor\ B3 = b, b_2, b_3, b_4\ \dots, b_{N^2}$
$K = k_1, k_2, .., k_t, k_1, k_2, \dots, k_{N^2}$

STEP VI : Mengoperasikan $Vektor\ R, G\ dan\ B$ dengan K menggunakan operasi XOR sehingga diperoleh vektor $R_2, G_2\ dan\ B_2$

STEP VII : Mengubah vektor $R_2, G_2\ dan\ B_2$ menjadi matriks dengan ordo yang sesuai *cipher-image*, sehingga diperoleh matriks $R, G\ dan\ B$.

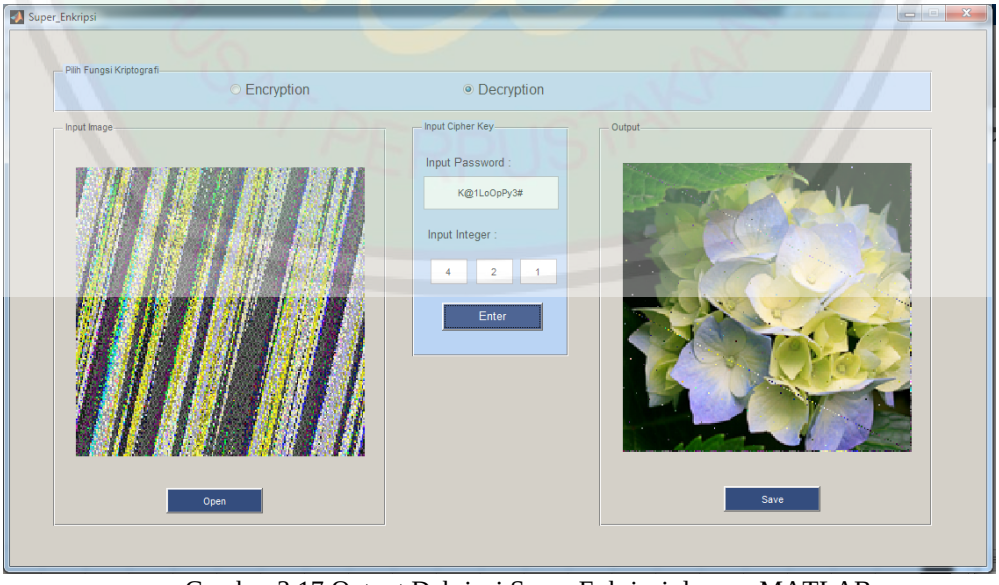
STEP VIII: Menggabungkan matriks $R, G\ dan\ B$ sehingga membentuk *plain-image.*

Baik proses enkripsi dan dekripsi dikemas dalam satu program, maka *interface* dalam proses dekripsi sama dengan gambar 3.7. selanjutnya untuk melakukan dekripsi pilih fungsi kriptografi *Decryption* dan menyertakan *cipher-image* serta *cipher-key* yang sesuai seperti pada gambar 3.16 berikut.



Gambar 3.16 Dekripsi citra dengan MATLAB

Langkah selanjutnya adalah menekan enter, sehingga gambar akan didekripsi sesuai *cipher-key* yang diberikan. Hasil citra pada *output* gambar 3.19 berupa *plain-image.* *Plain-image* tersebut merupakan citra asli dari bentuk *cipher*-image yang dibentuk menggunakan *password* K@1LoOpPy3# dan *integer* $b = 4, c = 2$ dan iterasi sebanyak 1.



Gambar 3.17 Output Dekripsi Super Enkripsi dengan MATLAB

3.5 Analisi Hasil Enkripsi Citra dengan MATLAB

3.5.1 Analisis Hasil Enkripsi dengan aplikasi MATLAB

Pada subbab ini simulasi akan dilakukan dengan menggunakan aplikasi MATLAB. Untuk membandingkan akurasi algoritma enkripsi maka digunakan dua contoh, yaitu citra warna RGB dan citra *grayscale* .

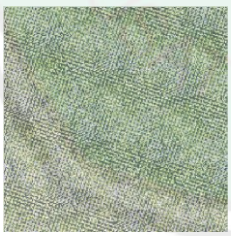
Gambar 3.18 Citra warna RGB

Gambar 3.19 Citra *grayscale*

Citra 3.18 dan 3.19 merupakan citra asli yang akan digunakan dalam proses enkripsi. Citra ini berukuran 200×200 pixel.

Berikut ini adalah hasil *cipher-image* perbandingan dari kunci enkripsi *Arnold Cat Map*.

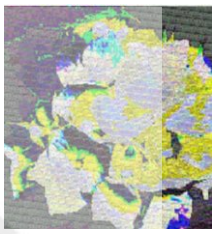
Tabel 3.2 Enkripsi *plain-image* dengan *Arnold Cat Map*

a=6, b=6, iterasi=30	a=1, b=1, iterasi=1	a=6, b=6, iterasi=1
		

Hasil enkripsi berbeda tergantung dengan kunci yang digunakan dan iterasi yang digunakan. Iterasi yang semakin besar akan menghasilkan *cipher-image* yang lebih baik dibandingkan dengan iterasi yang bernilai kecil.

Berikut ini adalah hasil *cipher-image* perbandingan dari kunci enkripsi *Vinegere Cipher* modifikasi *XOR*.

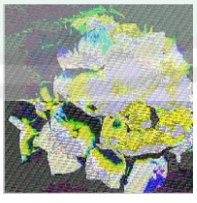
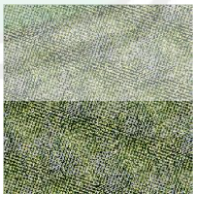
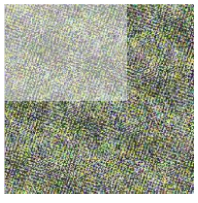
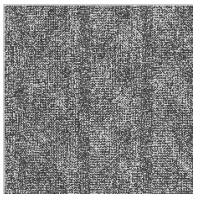
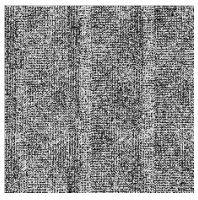
Tabel 3.3 Enkripsi *plain-image* berwarna dan *grayscale* dengan *Vinegere Cipher* modifikasi *XOR*

UNPREDICTABLE		Tidak DAPAT diprediksi dan ditentukan oleh @1230()	
			

Hasil enkripsi menggunakan *vinegere cipher* memangtak sebaik dengan Arnold cat map, karena algoritma ini hanya akan mengubah nilai pixel dan itu tergantung pada bervariasinya karakter pada kunci. Semakin bervariasi maka hasil enkripsi akan semakin baik.

Selanjutnya hasil enkripsi algoritma Super Enkripsi dengan kunci algoritma *Vinegere cipher* = AbcHyu36746ABCDEFGHIJK2 dan dilanjutkan dengan algoritma *Arnold cat map* dengan kunci $b = 6, c = 6$ dan iterasi $d = 50$.

Tabel 3.4 Enkripsi *plain-image* berwarna dan *grayscale* dengan Super Enkripsi

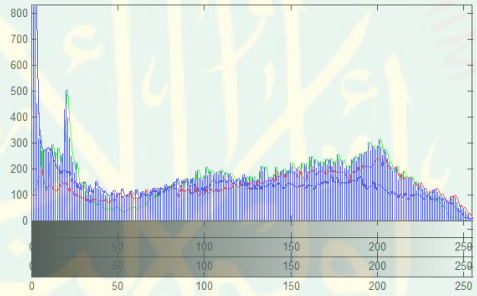
Citra	***Vinegere Cipher classic***	***Vinegere cipher* modifikasi XOR**	***Arnold Cat Map***	**Super Enkripsi**
				
				

Analisi hasil enkripsi pada tabel 3.4 ini akan di analisis pada subbab 3.5.2.

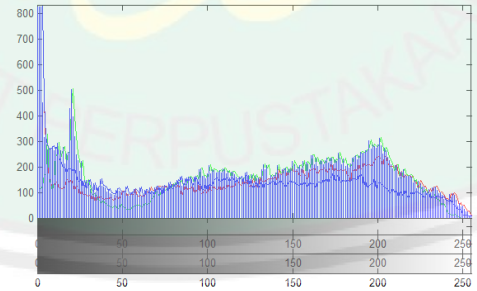
3.5.2 Analisis Histogram Citra Hasil enkripsi

Histogram dapat digunakan untuk mencari citra yang memiliki komposisi warna sama. Teknik ini dillakukan dengan melihat penyebaran warna pada histogram citra, selanjutnya dibandingkan histogram *plain-image* dengan *cipher-image* .Jika nilai histogram *cipher-image* memiliki distribusi keragaman dan perbedaaan yang signifikan dengan histogram *plain-image*, dapat dikatakan bahwa *cipher-image* tidak memberikan petunjuk apapun untuk melakukan *statistical attack* pada algoritma enkripsi yang digunakan.

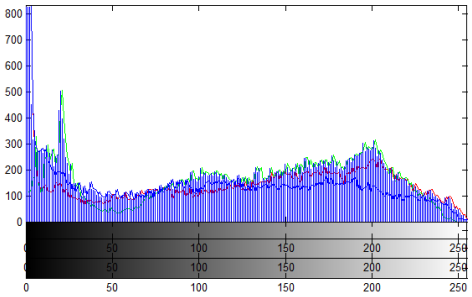
Berikut ini hasil analisis histogram dari setiap citra yang ada pada tabel 3.4.



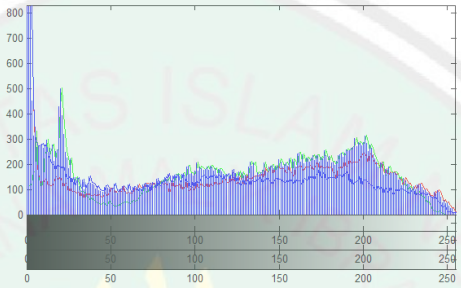
Gambar 3.20 Histogram *plain-image* citra warna pada tabel 3.3.



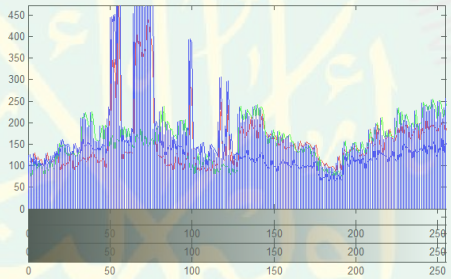
Gambar 3.21 Histogram *cipher-image* citra warna pada tabel 3.4 dengan *Vinegere Cipher Classic*



Gambar 3. 22 Histogram *cipher-image* citra warna pada tabel 3.4 dengan *Vinegere cipher* modifikasi XOR



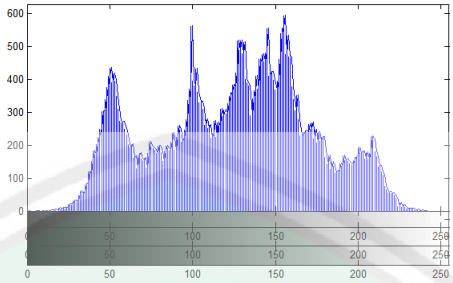
Gambar 3.23 Histogram *cipher-image* citra warna pada tabel 3.4 dengan *Arnold Cat Map*



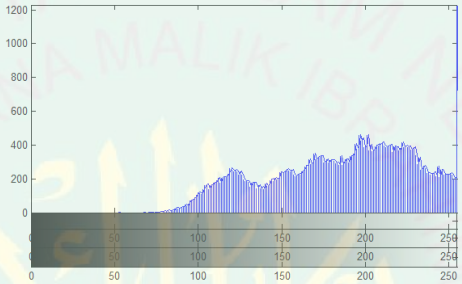
Gambar 3.24 Histogram *cipher-image* citra warna pada tabel 3.4 dengan Super Enkripsi

Gambar 3.20 s.d 3.24 merupakan hasil histogram dari gambar pada table 3.4 baris pertama berturut-turut pada kolom pertama sampai dengan kolom kelima. Gambar 3.20 merupakan histogram dari citra asli *(plain-image),* hasil histogram mempunyai gambaran yang sama dengan gambar 3.21, 3.22 dan 3.23 yang ketiganya merupakan histogram dari cipher-image hasil enkripsi dengan metode yang berbeda yaitu *vinegere cipher classic, vinegere cipher* modifikasi *XOR* dan *Arnold cat map.* Sedangkan gambar 3.20 mempunyai hasil yang berbeda dengan gambar 3.24 yang merupakan *cipher –image* dari metode Super Enkripsi. Maka dapat disimpulkan jika gambar berwarna yang di enkripsi dengan

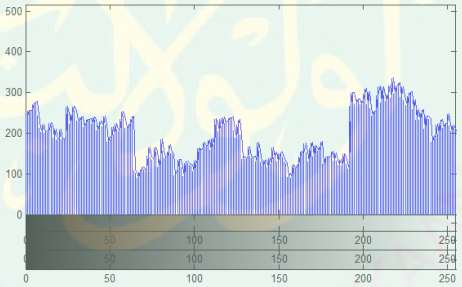
metode Super enkripsi lebih aman untuk terhindar dadi *statistical attack*, karena histogramnya jauh berbeda dengan histogram *plain-image*nya.



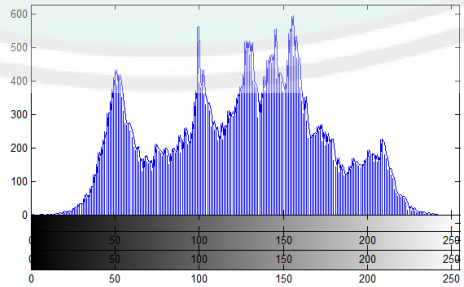
Gambar 3.25 Histogram *plain-image* citra *grayscale* pada tabel 3.4



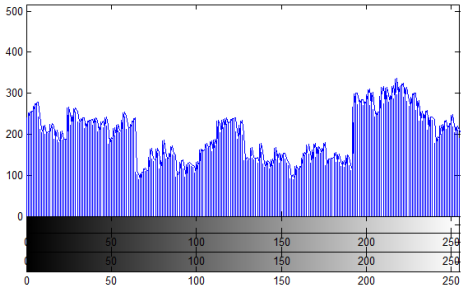
Gambar 3.26 Histogram *cipher-image* citra *grayscale* pada tabel 3.4 dengan *Vinegere Cipher Classic*



Gambar 3.27 Histogram *cipher-image* citra *grayscale* pada tabel 3.4 dengan *Vinegere Cipher* modifikasi XOR



Gambar 3.28 Histogram *cipher-image* citra *grayscale* pada tabel 3.4 dengan *Arnold Cat Map*



Gambar 3.29 Histogram *cipher-image* citra *grayscale* pada tabel 3.4 dengan Super Enkripsi.

Gambar 3.25 s.d 3.29 merupakan gambar histogram hasil enkripsi pada tabel 3.4 dibaris ke 2 kolom pertama sampai kelima secara berurutan. Gambar 3.25 merupakan histogram *plain-image* yang merupakan citra *grayscale.* Histogram ini mempunyai gambaran yang sama dengan gambar 3.28 yang merupakan histogram dari *cipher-image* dengan metode *Arnold cat map*. Sedangkan pada gambar 3.26, 3.27 dan 3.29 mempunyai hasil histogram yang jauh berbeda dengan *plain-image.* Sehingga dapat disimpulkan jika gambar *grayscale* jika dia enkripsi dengan *vinegere cipher classic, vinegere cipher* modifikasi *XOR* dan Super enkripsi akan menghasilkan *cipher-image* yang aman dari *stastical attack* karena mempunyai histogram yang berbeda dari *plain-image*nya.

Dari analisis hasil enkripsi baik pada gambar berwarna maupun gambar *grayscale* bisa dilihat jika super enkripsi lebih baik dalam menghasilkan hasil enkripsi baik dari segi hasil gambar yang tertera pada tabel 3.4 maupun hasil histogram. Metode lain baik *vinegere cipher classic* dan *vinegere cipher* modifikasi *XOR* hanya menghasilkan histogram yang berbeda dari *plain-image* jika citra yang digunakan tipe *grayscale*, dan juga hasil enkripsi gambarnya dapat dilihat secara kasat mata hampir mirip dengan *plain-image*nya baik pada citra *grayscale* maupun berwarna. Sedangkan *Arnold cat map* menghasilkan *cipher-image* yang cukup baik, namun hasil histogram pada *cipher-image*nya baik pada citra *grayscale* maupun berwarna mempunyai gambaran histogram yang sama dengan *plain-image*nya. Hal ini menyebabkan memudahkan pihak lain mengidentifikasi gambar secara statistik.

3.6 Kajian Keagamaan

Penyandian data atau pesan yang biasanya disebut proses enkripsi, bertujuan untuk melindungi pesan atau data dari pihak yang tidak mempunyai hak mengetahui isi pesan maupun data tersebut. Kerahasian suatu pesan gambar maupun teks pasti memiliki alasan dan tujuan tersendiri. Adapun salah satu tujuannya untuk melindungi privasi seseorang yang bisa jadi tidak baik untuk diketahui pihak lain ataupun masyarakat luas. Dalam Hadist yang diriwayatkan oleh Tirmidzdi, beliau menjelaskan tentang kebaikan yang akan didapatkan oleh seseorang yang mampu menutupi keburukan atau aib orang lain.

مَنْ نَفَّسَ عَنْ مُسْلِمٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا نَفَّسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ الْقِيَامَةِ وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ فِي الدُّنْيَا يَسَّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ وَمَنْ سَتَرَ عَلَى مُسْلِمٍ فِي الدُّنْيَا سَتَرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ وَاللَّهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيهِ

Artinya:

"Barangsiapa yang meringankan (menghilangkan) kesulitan seorang muslim kesulitan-kesulitan duniawi, maka Allah akan meringankan (menghilangkan) baginya kesulitan di akhirat kelak. Barangsiapa yang memberikan kemudahan bagi orang yang mengalami kesulitan di dunia, maka Allah akan memudahkan baginya kemudahan (urusan) di dunia dan akhirat. ***Dan* barangsiapa yang menutupi (aib) seorang muslim *sewaktu di dunia, maka Allah akan menutup (aibnya) di dunia dan akhirat.*** *Sesungguhnya Allah akan senantiasa menolong seorang hamba selalu ia menolong saudaranya*." [HR. Tirmidzi]

Dalam hadist diatas telah dijelaskan bahwa keburukan seseorang hendaklah ditutupi dan jangan sampai membaginya kepada orang lain,karena Allah SWT juga akan menutupi keburukan dari seseorang itu. Jika seseorang mau menutupi keburukan dari saudara ataupun orang lain maka balasan yang ia dapat dari Allah SWT adalah kebaikan juga.

Selain untuk menjaga privasi orang lain, mengenkripsi pesan juga bertujuan untuk menjaga amanah yang telah diberikan kepadanya sehigga tidak tersebar pada pihak lain yang tidak berwenang. Pada surat Al-Anfal ayat 27

berikut ini telah disampaikan larangan dalam mengkhianatai amanah yang telah diberikan.

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُواْ لَا تَخُونُواْ ٱللَّهَ وَٱلرَّسُولَ وَتَخُونُوٓاْ أَمَٰنَٰتِكُم وَأَنتُم تَعلَمُونَ

Artinya :
Hai orang-orang yang beriman, janganlah kamu mengkhianati Allah dan Rasul (Muhammad) dan (juga) janganlah kamu mengkhianati amanat-amanat yang dipercayakan kepadamu, sedang kamu mengetahui (QS. Al-Anfal/8:27)

Adapun ayat Al-Quran yang juga mendukung amanat yang dikandung dalam Al-Quran surat Al-Anfal ini yaitu Al-Qur'an surat An-nisa ayat 58 yang berbunyi:

إِنَّ ٱللَّهَ يَأمُرُكُم أَن تُؤَدُّواْ ٱلأَمَٰنَٰتِ إِلَىٰٓ أَهلِهَا وَإِذَا حَكَمتُم بَينَ ٱلنَّاسِ أَن تَحكُمُواْ بِٱلعَدلِۚ إِنَّ ٱللَّهَ نِعِمَّا يَعِظُكُم بِهِۦٓۗ إِنَّ ٱللَّهَ كَانَ سَمِيعَۢا بَصِيرا

Artinya :
Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha mendengar lagi Maha melihat (QS. An-Nisa/4:58)

Amanah memiliki arti dipercaya atau terpercaya, amanah berkaitan tentang menjaga kepercayaan orang lain atas sesuatu yang telah dititipkan oleh pemberi amanah. Hal ini bisa diartikan amanah dapat berupa benda, perkataan, perbuatan maupun pesan. Kedua ayat tersebut mempunyai pesan bahwa amanah-amanah yang telah diberikan haruslah disampaikan kepada yang berhak menerima. Dan mereka yang mau menjaga amanah adalah termasuk orang-orang yang beriman. Maka bagi mereka yang bisa menjaga kepercayaan maka ia adalah orang yang beriman.

BAB IV

PENUTUP

4.1 **Kesimpulan**

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa dengan menggunakan super enkripsi dengan *Vinegere cipher* dan *Arnold Cat Map* didapatkan keamanan ganda. Pertama keamanan terletak pada nilai setiap pixel yang berubah sesuai kunci berupa *password* yang dimasukkan. Kunci password yang digunakan dalam *Vinegere cipher* akan diektrak dalam kode ASCII, ada 255 kode (jumlah ASCII kode). Jika *Vinegere cipher* menggunakan kunci minimal 3 karakter, namun semakin banyak karakter yang digunakan maka gambar akan semakin terenkripsi dengan nbaik. Maka semakin panjang kunci yang digunakan maka untuk memecahkan kuncinya akan semakin sulit. Selanjutnya keamanan kedua pixel gambar akan tersebar sesuai dengan transformasi yang dilakukan oleh *Arnold Cat Map (ACM)*.

Pada proses dekripsinya menggunakan kunci yang sama dengan dekripsinya. Proses dekripsi akan dimulai dengan dekripsi algoritma *Arnlod Cat Map*. Ada tiga kombinasi kunci *integer* yang digunakan pada *ACM* ini, jika salah satu integer salah maka kunci tidak dapat mengembalikan kebentuk semula. Kemudian dekripsi dilanjutkan dengan algoritma *Vinegere cipher*, kunci yang digunakan berupa karakter baik huruf maupun symbol. Kombinasi kunci yang digunakan juga dapat bervariasi , semakin bervariasi kunci maka akan semakin membagus hasil enkripsi. jika ada yang bisa memecahkan salah satu kunci

algoritma baik milik *vinegere cipher* atau *Arnold Cat Map,* hal ini juga tidak akan mengembalikan citra pada bentuk aslinya.

4.2 Saran

Pada penelitian ini membahas mengenai super enkripsi yang menggabungkan algoritma *Vinegere cipher* dan *Arnold Cat Map* pada Matriks citra . Kunci yang digunakan pada penelitian ini merupakan kunci simetri. Untuk pengembangan Penelitian selanjutnya disarankan menggunakan metode asimetri maupun gabungan kunci simetri dengan asimetri atau merode lain untuk memperkuat pengamanan pesan berupa citra.

DAFTAR RUJUKAN

Andrianto, H. dan Prijo, A. 2006. *Menguasai Matriks dan Vektor*. Bandung: Rekayasa Sains.

Anton, H dan Rorres, C. 2010. *Alementary Linier Algebra Aplications Version.* New Jersey : John Wiley & Sons, ets.

Anton, Howard. 1997. *Aljabar Elementer, Edisi Kelima*. Jakarta: Erlangga.

Ariyus, Doni. 2006. *Kriptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Binanto, Iwan. 2010. *Multimedia Digital Dasar Teori + Pengembangannya*. Yogyakarta: Andi Offset.

Gonzales, Rafael . C. dan Richard E. Woods. 2002. *Digital Image Processing.* New Jersey : Prentice Hall.

McAndrew,Alasdair. 2011. *Introduction To Cryptography with Open-Soutce Software.* Melbourne: CRC Press.

Munir, Rinaldi. 2012. Algoritma Enkripsi Citra Digital Berbasis Chaos Dengan Penggabungan Teknik Permutasi Dan Teknik Substitusi Menggunakan Arnold Cat Map Dan Logistic Map. *ISSN* 2087-2658.

Nafi'iyah, Nur. 2015. Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner. *Jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA)* Vol.9, No.2.

Prakoso, Sigit Buddy dan Ardi Pujiyanta. 2014. Media Pembelajaran perhitungan Determinan Reduksi Minor Ekspansi Kofaktor dan Adjoin. Jurnal Sarjana teknik Informatika Vol. 2 No.1.

Purba, R. Arwin Halim, Indra Syahputra. 2014. Enkripsi Citra Digital Menggunakan *Arnold's Cat Map* dan *Nonlinear Chaotic Algorithm. ISSN*. 1412-0100.

Purwanto, H., Indriani. G., dan Dayanti,E. 2005. *Aljabar Linier*. Jakarta : PT. Ercontara Rajawali.

Stallings, William. 2003. *Cryptography and Network Security*. New Jersey: Pearson Education.

Suryadi, Zuherman Rustam, Wiwit Widhianto.2014. Implementasi Algoritma Enkripsi Citra Digital Menggunakan Skema Tranposisi Berbasis Fungsi Chaos. *ISSN*: 2302-3740.

Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, Oky D., dan Wijanarto. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Andi Offset.

LAMPIRAN

```
function varargout = Super_Enkripsi_coba1(varargin)
% SUPER_ENKRIPSI_COBA1 MATLAB code for
Super_Enkripsi_coba1.fig
%      SUPER_ENKRIPSI_COBA1, by itself, creates a new
SUPER_ENKRIPSI_COBA1 or raises the existing
%      singleton*.
%
%      H = SUPER_ENKRIPSI_COBA1 returns the handle to a
new SUPER_ENKRIPSI_COBA1 or the handle to
%      the existing singleton*.
%
%
SUPER_ENKRIPSI_COBA1('CALLBACK',hObject,eventData,handle
s,...) calls the local
%      function named CALLBACK in SUPER_ENKRIPSI_COBA1.M
with the given input arguments.
%
%      SUPER_ENKRIPSI_COBA1('Property','Value',...)
creates a new SUPER_ENKRIPSI_COBA1 or raises the
%      existing singleton*.  Starting from the left,
property value pairs are
%      applied to the GUI before
Super_Enkripsi_coba1_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to
Super_Enkripsi_coba1_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose
"GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Super_Enkripsi_coba1

% Last Modified by GUIDE v2.5 17-Dec-2018 15:07:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@Super_Enkripsi_coba1_OpeningFcn, ...
                   'gui_OutputFcn',
@Super_Enkripsi_coba1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Super_Enkripsi_coba1 is made
visible.
function Super_Enkripsi_coba1_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to
Super_Enkripsi_coba1 (see VARARGIN)

% Choose default command line output for
Super_Enkripsi_coba1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Super_Enkripsi_coba1 wait for user
response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the
command line.
function varargout =
Super_Enkripsi_coba1_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata,
handles)
set(handles.radiobutton1,'value',0)
pilih=2;
handles.pilih=pilih;
```

```
guidata(hObject,handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton2


% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata,
handles)
set(handles.radiobutton2,'value',0)
pilih=1;
handles.pilih=pilih;
guidata(hObject,handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton1



function password_Callback(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
password as text
%        str2double(get(hObject,'String')) returns
contents of password as a double


% --- Executes during object creation, after setting all
properties.
function password_CreateFcn(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end



function integer1_Callback(hObject, eventdata, handles)
% hObject    handle to integer1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
integer1 as text
%        str2double(get(hObject,'String')) returns
contents of integer1 as a double


% --- Executes during object creation, after setting all
properties.
function integer1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to integer1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function integer2_Callback(hObject, eventdata, handles)
% hObject    handle to integer2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
integer2 as text
%        str2double(get(hObject,'String')) returns
contents of integer2 as a double


% --- Executes during object creation, after setting all
properties.
function integer2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to integer2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function integer3_Callback(hObject, eventdata, handles)
% hObject    handle to integer3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
integer3 as text
%        str2double(get(hObject,'String')) returns
contents of integer3 as a double



% --- Executes during object creation, after setting all
properties.
function integer3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to integer3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in enter.
function enter_Callback(hObject, eventdata, handles)
integer_1=str2num(get(handles.integer1,'string'))
integer_2=str2num(get(handles.integer2,'string'))
integer_3=str2num(get(handles.integer3,'string'))
kunci=get(handles.password,'string')
ASCI=['
',''','"','#','$','%','&','\','(',')','*','+',',','-
','.','/','0','1','2','3','4','5','6','7','8','9',':',';
','<','=','>','?','@','A','B','C','D','E','F','G','H','I
','J','K','L','M','N','O','P','Q','R','S','T','U','V','W
','X','Y','Z','[','\',']','^','-
','`','a','b','c','d','e','f','g','h','i','j','k','l','m
```

```
','n','o','p','q','r','s','t','u','v','w','x','y','z','{
','|','}','~']

pilih=handles.pilih;
switch handles.pilih
    case 1
        gambar=getimage(handles.axes2);
        size_g=size(gambar);
        panjang_g=size_g(2);
        lebar_g=size_g(1);

        r=gambar(:,:,1);
        g=gambar(:,:,2);
        b=gambar(:,:,3);

        size_p=size(gambar);
        baris=size_p(1);
        kolom=size_p(2);
        panjang_k=length(kunci);
        panjang_p=baris*kolom;

%mentrasformasikan layer r kedalam array
        trans_r=transpose(r);
        panjang_r=baris*kolom;
        layer_r=reshape(trans_r, 1, panjang_r);

%mentrasformasikan layer g kedalam array
        trans_g=transpose(g);
        panjang_g=baris*kolom;
        layer_g=reshape(trans_g, 1, panjang_g)

%mentrasformasikan layer b kedalam array
        trans_b=transpose(b);
        panjang_b=baris*kolom;
        layer_b=reshape(trans_b, 1, panjang_b)

%memanjangkan kunci sebanyak array plain-image
if panjang_k < panjang_p
     keyReps = floor(panjang_p/panjang_k);
     extendedKey = [repmat(kunci,1,keyReps)
kunci(1:rem(panjang_p,panjang_k))];
 else
     extendedKey = kunci(1:panjang_p);
end

 %Merubah kunci kedalam bentuk bilangan dengan code
ASCII
 for i=1:panjang_p
    vec1(i)=find(ASCI==(extendedKey(i)))+31;
 end

 %Melakukan persamaan Vinegere cipher  pada layer r
 hasil_r=bitxor(uint8(vec1),uint8(layer_r));

 %Melakukan persamaan Vinegere cipher  pada layer g
  hasil_g=bitxor(uint8(vec1),uint8(layer_g));
```

```
%Melakukan persamaan Vinegere cipher  pada layer b
 hasil_b=bitxor(uint8(vec1),uint8(layer_b));

 for i=1
   for j=1:panjang_p
       hasil_red(i,j)=mod(layer_r(i,j)+ vec1(i,j),256);
   end
end

%Melakukan persamaan Vinegere cipher  pada layer g
for i=1
   for j=1:panjang_p
       hasil_green(i,j)=mod(layer_g(i,j)+
vec1(i,j),256);
   end
end

%Melakukan persamaan Vinegere cipher  pada layer b
for i=1
   for j=1:panjang_p
       hasil_blue(i,j)=mod(layer_b(i,j)+
vec1(i,j),256);
   end
end


%merubah array hasil_r hasil ke dalam matrik awal
reshape_r=reshape(hasil_r, baris, kolom);
hasillayer_r=transpose(reshape_r);

 %merubah array hasil_g hasil ke dalam matrik awal
reshape_g=reshape(hasil_g, baris, kolom);
hasillayer_g=transpose(reshape_g);

%merubah array hasil_g hasil ke dalam matrik awal
reshape_b=reshape(hasil_b, baris, kolom);
hasillayer_b=transpose(reshape_b);

semua=cat(3,hasillayer_r, hasillayer_g, hasillayer_b);


ima=semua;
r=ima(:,:,1);
g=ima(:,:,2);
blue=ima(:,:,3);
u=size(ima);
i=u(1);
j=u(2);
temp=1;
ite=integer_3;
b=integer_1
c=integer_2
M=[1 b; c b*c+1];
while( temp<=ite)
     for x=1:i
         for y=1:j
             hasil=mod(M*[x;y],i);
```

```
                row=hasil(1);
                col=hasil(2);
                baris=mod(row,i);
                kolom=mod(col,j);
                if baris==0;
                    b=baris+i;
                elseif kolom==0;
                    ko=kolom+j;
                 else
                     b=baris;
                     ko=kolom;
                end
                red(x,y)=r(b,ko);
                green(x,y)=g(b,ko);
                bluee(x,y)=blue(b,ko);
            end
        end
        temp=temp+1;
        r=red;
        g=green;
        blue=bluee;
end
    all=cat(3,r,g,blue);

 handles.all=all; %menyimpan nilai variable
    guidata(hObject,handles);
    axes(handles.axes3);
    imshow(all);

    case 2
ima=getimage(handles.axes2);
r=ima(:,:,1);
g=ima(:,:,2);
blue=ima(:,:,3);
u=size(ima);
i=u(1);
j=u(2);
temp=1;
ite=integer_3;
b=integer_1
c=integer_2
M=[b*c+1 -1*b; -1*c 1];
while( temp<=ite)
     for x=1:i
         for y=1:j
                hasil=mod(M*[x;y],i);
                row=hasil(1);
                col=hasil(2);
                baris=mod(row,i);
                kolom=mod(col,j);
                if baris==0;
                    b=baris+i;
                elseif kolom==0;
                    ko=kolom+j;
                 else
                     b=baris;
                     ko=kolom;
                end
```

```
                red(x,y)=r(b,ko);
                green(x,y)=g(b,ko);
                bluee(x,y)=blue(b,ko);
            end
        end
        temp=temp+1;
        r=red;
        g=green;
        blue=bluee;
end
    all_d=cat(3,r,g,blue);

    %Dekrip Vinegere
    gambar_d=all_d;
    size_g_d=size(gambar_d);
    panjang_g_d=size_g_d(2);
    lebar_g_d=size_g_d(1);

r_d=gambar_d(:,:,1);
g_d=gambar_d(:,:,2);
b_d=gambar_d(:,:,3);

size_p_d=size(gambar_d);
baris_d=size_p_d(1);
kolom_d=size_p_d(2);
panjang_k_d=length(kunci);
panjang_p_d=baris_d*kolom_d;

%mentrasformasikan layer r kedalam array
trans_r_d=transpose(r_d);
panjang_r_d=baris_d*kolom_d;
layer_r_d=reshape(trans_r_d, 1, panjang_r_d);

%mentrasformasikan layer g kedalam array
trans_g_d=transpose(g_d);
panjang_g_d=baris_d*kolom_d;
layer_g_d=reshape(trans_g_d, 1, panjang_g_d);

%mentrasformasikan layer b kedalam array
trans_b_d=transpose(b_d);
panjang_b_d=baris_d*kolom_d;
layer_b_d=reshape(trans_b_d, 1, panjang_b_d);

%memanjangkan kunci sebanyak array plain-image
if panjang_k_d < panjang_p_d
     keyReps_d = floor(panjang_p_d/panjang_k_d);
     extendedKey_d = [repmat(kunci,1,keyReps_d)
kunci(1:rem(panjang_p_d,panjang_k_d))];
 else
     extendedKey_d = kunci(1:panjang_p_d);
end

 %Merubah kunci kedalam bentuk bilangan dengan code
ASCII
 for i=1:panjang_p_d
    vec1_d(i)=find(ASCI==(extendedKey_d(i)))+31;
 end
```

```
%Melakukan persamaan Vinegere cipher  pada layer r
hasil_r_d=bitxor(uint8(vec1_d),uint8(layer_r_d));

%Melakukan persamaan Vinegere cipher  pada layer g
 hasil_g_d=bitxor(uint8(vec1_d),uint8(layer_g_d));

%Melakukan persamaan Vinegere cipher  pada layer b
 hasil_b_d=bitxor(uint8(vec1_d),uint8(layer_b_d));

%merubah array hasil_r hasil ke dalam matrik awal
reshape_r_d=reshape(hasil_r_d, baris_d, kolom_d);
hasillayer_r_d=transpose(reshape_r_d);

 %merubah array hasil_g hasil ke dalam matrik awal
reshape_g_d=reshape(hasil_g_d, baris_d, kolom_d);
hasillayer_g_d=transpose(reshape_g_d);

%merubah array hasil_g hasil ke dalam matrik awal
reshape_b_d=reshape(hasil_b_d, baris_d, kolom_d);
hasillayer_b_d=transpose(reshape_b_d);

 alldy=cat(3,hasillayer_r_d, hasillayer_g_d,
hasillayer_b_d);

    handles.alldy=alldy; %menyimpan nilai variable
    guidata(hObject,handles);
    axes(handles.axes3);
    imshow(alldy);

end

% hObject    handle to enter (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
all=handles.all;
[name_file_save,path_save] = uiputfile( ...
    {'*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File jpeg (*.jpg)';
    '*.tif','File Tif (*.tif)';
    '*.*','All Files (*.*)'},...
    'Save Image');
if ~isequal(name_file_save,0)
    imwrite(all,fullfile(path_save,name_file_save));
else
    return
end
```

```
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)


% --- Executes on button press in open.
function Open(hObject, eventdata, handles)
% hObject    handle to open (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)


% --- Executes on button press in open.
function open_Callback(hObject, eventdata, handles)
[FileName,PathName]=uigetfile(...
    {'*.bmp;*.jpg;*.tif','Files of type
(*.bmp,*.jpg,*.tif)';
    '*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File jpeg (*.jpg)';
    '*.tif','File Tif (*.tif)';
    '*.*','All Files (*.*)'},...
    'Select Image File')
im=imread([PathName,FileName]);
handles.im=im; %menyimpan nilai variable
guidata(hObject,handles);
axes(handles.axes2);
imshow(im);

% hObject    handle to open (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)


% --- Executes during object creation, after setting all
properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
```

RIWAYAT HIDUP

Cici Erisa Maulidah, lahir di Pasuruan pada tanggal 16 Agustus 1996. biasa dipanggil Eris oleh keluarga tapi kebanyakan teman memanggil Cici. Ia anak pertama dari 2 bersaudara pasangan bapak Sugeng Hadi P. dan Ibu Khotima Tusifak dan merupakan Kakak dari Kiki Nur Alvin .

Pendidikan dasarnya ditempuh di SDN 1 Sekarmojo dan lulus pada tahun 2008. Setelah itu melanjutkan sekolah di SMPN 1 Purwosari, lulus tahun 2011. Pendidikan selanjutnya ditempuh di SMKN 1 Purwosari dalam bidang Multimedia. Dan telah mengikuti bergabagi kegiatan dalam masa sekolahnya seperti Pecinta Alam dan Jurnalistik, kemudian lulus tahun 2014. Selanjutnya, pada tahun yang sama melanjutkan kuliah di Universitas Islam Negeri Maulana Malik Ibrahim Malang Jurusan Matematika.

Selama menjadi mahasiswa telah mengikuti beberapa kegiatan ia lakukan seperti mengikuti organisasi Korp Sukarela dan bisnis jasa *design Vector*.

KEMENTRIAN AGAMA RI
UINIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No. 50 Dinoyo Malang Telp/Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Cici Erisa Maulidah
NIM :14610059
Fakultas/Jurusan : Sains dan Teknologi/ Matematika
Judul Skripsi : Implementasi Metode Super Enkripsi (*Vinegere cipher –Arnold Cat Map*) pada Matriks Citra
Pembimbing I : Dr. H. Turmudi, M.Si, Ph. D
Pemimbing II : M. Khudzaifah, M. Si

No.	Tanggal	Hal	Tanda Tangan	
1	27 April 2018	Konsultasi Bab 1	1.	
2	7 Mei 2018	Konsultasi Bab 1 dan Keagamaan		2.
3	6 Juli 2018	Konsultasi Bab II	3.	
4	9 Juli 2018	Konsultasi Bab II dan Bab III		4.
5	6 Agustus 2018	Konsultasi Bab III	5.	
6	29 September 2018	Konsultasi Bab III		6.
7	12 Oktober 2018	Konsultasi Bab III	7.	
8	19 Oktober2018	Konsultasi Bab III		8.
9	30 Oktober 2018	Konsultasi Bab I , II, III dan IV	9.	
10	1 November2018	Konsultasi Bab I , II, III dan IV		10.
11	7 November 2018	Konsultasi Bab I , II, III dan IV	11.	

Malang, 11 November 2018
Mengetahui,
Ketua Jurusan

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 001