BAB II
TINJAUAN PUSTAKA

2.1 Gambar

Gambar adalah penyampaian informasi dalam bentuk visual. Penyampaian informasi dalam bentuk gambar akan lebih menarik dan efektif, karena gambar dapat meringkas data yang komplek dengan cara yang baru dan lebih berguna. *Image* pada komputer merupakan *array* bilangan yang merepresentasikan nilai intensitas cahaya yang bervariasi (*pixel*). Kumpulan *pixel-pixel* inilah yang membentuk suatu gambar. Gambar yang sering digunakan adalah 24 *bits* dan gambar 8 *bits*(256 *colors*). Pada *steganografi*, gambar yang biasa digunakan adalah 24 bits, karena gambar 24 *bits* tersebut dapat menyediakan *space* yang besar untuk disisipi oleh data. Piksel penyusun gambar ini tersusun atas 3 warna primer yaitu merah, hijau, dan biru. Masing – masing primer tersusun atas 1 *byte* data. Untuk image 24 *bits* berarti menggunakan 3 *bytes* per piksel untuk merepresentasikan nilai warna *pixel*. 3 *bytes* data ini dapat berupa heksadesimal, desimal, ataupun biner. Pada kebanyakan webpage menggunakan warna *background* yang direpresentassikan oleh 6 digit bilangan hekadesimal 000000 sampai FFFFFF yang terdiri atas 3 warna primer yang masing-masing tersusun atas 2 digit heksadesimal. Putih bernilai FFFFFF biru bernilai FF0000, hijau bernilai 00FF00, merah bernilai 0000FF, dan hitam bernilai 000000. Nilai dalam desimal berkisar antara 0 sampai 255 untuk masing-masing warna primer dan 00000000 sampai 11111111 untuk nilai dalam biner.

Berikut ini adalah penjelasan dari berbagai format gambar :

1. BMP (*Bitmap Image)*

   Format file ini merupakan format grafis yang fleksibel untuk platform Windows sehingga dapat dibaca oleh program grafis manapun.Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit. Kelemahan format file ini adalah tidak mampu menyimpan alpha channel serta ada kendala dalam pertukaran platform. Untuk membuat sebuah objek sebagai desktop wallpaper, simpanlah dokumen Anda dengan format file ini. Anda dapat mengkompres format file ini dengan kompresi RLE. Format file ini mampu menyimpan gambar dalam mode warna RGB, Grayscale, Indexed Color, dan Bitmap.

2. GIF (*Graphic Interchange Format)*

   Format file ini hanya mampu menyimpan dalam 8 bit (hanyamendukung mode warna *Grayscale, Bitmap dan Indexed Color*).Format file ini merupakan format standar untuk publikasi elektronik dan internet. Format file mampu menyimpan animasi dua dimensi yang

akan dipublikasikan pada internet, desain halaman web dan publikasi elektronik. Format file ini mampu mengkompres dengan ukuran kecil menggunakan kompresi LZW.

3. TIF (*Tagged Image Format File)*

   Format file ini mampu menyimpan gambar dengan kualitas hingga32 bit. Format file ini juga dapat digunakan untuk keperluan pertukaran antar *platform* (PC, Machintosh, dan Silicon Graphic). Format file ini merupakan salah satu format yang dipilih dan sangat disukai oleh para pengguna komputer grafis terutama yang

   berorientasi pada publikasi (cetak). Hampir semua program yangmampu membaca format file bitmap juga mampu membaca format file TIF.

4. PNG *(Portable Network Graphic)*

   Format file ini berfungsi sebagai alternatif lain dari format file GIF. Format file ini digunakan untuk menampilkan objek dalam halaman web. Kelebihan dari format file ini dibandingkan dengan GIF adalah kemampuannya menyimpan file dalam bit depth hingga 24 bit serta mampu menghasilkan latar belakang (*background*) yangtransparan dengan pinggiran yang halus. Format file ini mampu menyimpan alpha channel.

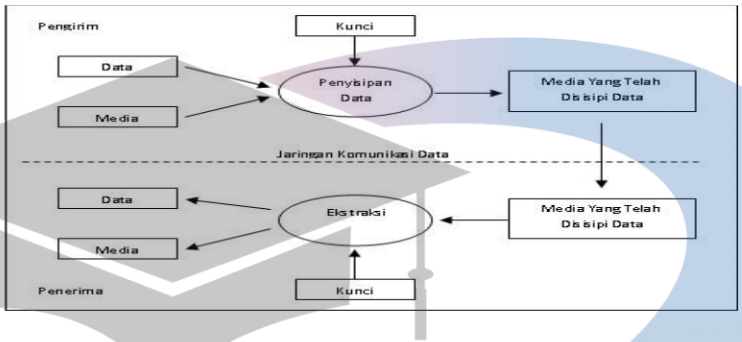
5. JPG/JPEG (*Joint Photographic Expert Group*)

   Format file ini mampu mengkompres objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan. Format file sering dimanfaatkan untuk menyimpan gambar yang akan digunakan untuk keperluan halaman web, multimedia, dan publikasielektronik lainnya. Format file ini mampu menyimpan gambar dengan mode warna RGB, CMYK, dan *Grayscale*. Format file ini juga mampu menyimpan *alpha channel*, namun karena orientasinya ke publikasi elektronik maka format ini berukuran relatif lebih kecil dibandingkan dengan format file lainnya.

2.2 Steganografi

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan pesan tersebut tidak terdeteksi oleh indera manusia (Munir,2004). Tujuan utama dari *steganografi* adalah untuk menyembunyikan data ke dalam data lainnya sehingga tidak memungkinkan pihak ketiga untuk mendeteksi keberadaan pesan yang disembunyikan. *Steganografi* memanfaatkan kekurangan-kekurangan sistem indera manusia seperti mata (*Human Visual System*) dan telinga (*Human Auditory System*), sehingga tidak diketahui kehadirannya oleh indera manusia (indera penglihatan atau indera pendengaran) dan mampu menghadapi proses-proses pengolahan sinyal digital dengan tidak merusak kualitas data yang telah disisipi sampai pada tahap tertentu. *Steganografi* dan

kriptografi memiliki tujuan yang sama yaitu mengamankan suatu informasi, namun cara pengamanannya berbeda. Jika pada kriptografi pesan diacak dalam bentuk pesan yang telah disandikan (*ciphertext*), sedangkan pada *steganografi* pesan disembunyikan sehingga tidak dapat dilihat oleh indera penglihatan. Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan.



Gambar 2. 1 Diagram Sistem Steganografi.

Pada gambar 2.1, bagian atas terlihat diagram proses penyisipan yang dilakukan oleh si pengirim yaitu data rahasia disembunyikan pada media penampung dengan menggunakan kunci. Pada bagian bawah, terlihat diagram proses ekstraksi oleh penerima untuk mendapatkan kembali data rahasia.

2.2.1 Karakteristik Steganografi

Menurut Munir (2004), ada beberapa kriteria yang harus diperhatikan dalam steganografi, yaitu :

1. *Imperceptibility*.

   Keberadaan pesan rahasia tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka indera telinga tidak dapat mendeteksi perubahan pada audio *stegotext*-nya.

2. *Fidelity*.

   Mutu *stegomedium* tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka *audio stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan tersebut.

3. *Recovery*.

Pesan yang disembunyikan harus dapat dikembalikan. Karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

2.2.2. Media Steganografi

Steganografi menggunakan sebuah berkas yang disebut dengan *cover*, tujuannya sebagai kamuflase dari pesan yang sebenarnya. Media yang sering digunakan pada steganografi adalah media digital seperti teks, gambar, *audio* dan *video*.

2.2.2.1 Steganografi Pada Teks

Teknik steganografi yang menggunakan teks sebagai *cover* adalah hal yang menantang. Ini dikarenakan berkas teks memiliki ukuran data yang kecil untuk bisa digantikan dengan berkas rahasia. Dan kekurangan lainnya adalah teks yang mengandung teknik steganografi ini dengan mudah dapat diubah oleh pihak yang tidak diinginkan dengan cara mengubah teks itu sendiri maupun mengubah format dari teksnya (misal .TXT menjadi .PDF). Ada beberapa metode yang digunakan pada media teks ini yaitu, *Line-Shift Encoding, Word-shift Encoding* dan *Feature Coding*. Ketiganya merupakan metode *encoding* yang membutuhkan berkas asli dan juga format aslinya untuk dapat di *decode* atau di ekstrak kembali.

2.2.2.2 Steganografi Pada Gambar

Steganografi pada gambar adalah metode yang paling banyak digunakan secara luas di dunia digital saat ini. Hal ini dikarenakan keterbatasan kemampuan dari visual atau *Human Visual System* (HVS). File gambar pada komputer merupakan array bilangan yang merepresentasikan nilai intensitas cahaya yang bervariasi (piksel). Kumpulan piksel-piksel inilah yang membentuk suatu gambar. Citra yang sering digunakan pada umumnya adalah citra 24 bit dan citra 8 bit (256 colors). Format citra yang biasanya digunakan adalah format *bitmap* (.bmp), *gif*, *jpeg* dan format citra lainnya. Hampir semua *plain* teks, *cipher* teks, gambar dan media lainnya dapat di *encode* kedalam aliran bit untuk disembunyikan didalam gambar digital. Pendekatan yang paling sering dilakukan pada media jenis ini adalah : *Least Significant Bit Insertion, Masking and Filtering* dan *Algorithm and Transformation*.

2.2.2.3 Steganografi Pada Audio

Penyembunyian data pada *audio* merupakan teknik yang paling menantang pada steganografi ini. Hal ini disebabkan *Human Auditory System* (HAS) memiliki jangkauan yang dinamis. HAS memiliki kemampuan mendengar lebih dari satu sampai 1 miliar. Dan jangkauan frekuensi lebih dari satu hingga seribu. *Auditory System* ini juga sangat peka pada gangguan

suara (*noise*) yang halus sekalipun. Sedikit saja terdapat gangguan pada sebuah berkas audio maka dengan mudah akan terdeteksi. Satu-satunya kelemahan yang dimiliki HAS dalam membedakan suara adalah kenyataan bahwa suara keras bisa menenggelamkan suara pelan. Terdapat dua konsep yang harus dipertimbangkan sebelum memilih metode mana yang akan dipakai, yaitu format digital audio dan media transmisi dari audio.

2.2.2.4 Steganografi Pada Video

Steganografi pada video merupakan penggabungan antara frame (banyak gambar) dan audio. Dibandingkan dengan media digital lainnya, media video mampu menampung pesan lebih banyak karena memiliki ukuran yang besar.

2.3 Teknik Steganografi

Menurut Sellars (2006), ada tujuh teknik dasar yang digunakan dalam steganografi, yaitu :

1. *Injection*

   Merupakan suatu teknik menanamkan pesan rahasia secara langsung ke suatu media. Salah satu masalah dari teknik ini adalah ukuran media yang diinjeksi menjadi lebih besar dari ukuran normalnya sehingga mudah dideteksi. Teknik ini sering juga disebut *embedding*.

2. Substitusi

   Data normal digantikan dengan data rahasia. Biasanya, hasilteknik ini tidak terlalu mengubah ukuran data asli, tetapi tergantung pada *file* media dan data yang akan disembunyikan. Teknik substitusi bisa menurunkan kualitas media yang ditumpangi.

3. *Transform Domain*

   Sebuah teknik yang sangat efektif. Pada dasarnya,transformasi domain menyembunyikan data pada *transform space*. Akan sangat lebih efektif teknik ini diterapkan pada *file* berekstensi JPG.

4. *Spread Spectrum*

   Sebuah teknik pengtransmisian menggunakan *pseudonoisecode*, yang independen terhadap data informasi sebagai modulator bentuk gelombang untuk menyebarkan energi sinyal dalam sebuah jalur komunikasi (*bandwidth*) yang lebih besar daripada sinyal jalur komunikasi informasi. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise code* tersinkronisasi.

5. *Statistical Method*

Teknik ini disebut juga skema *steganographic* 1 bit. Skema tersebut menanamkan satu bit informasi pada media tumpangan dan mengubah statistik walaupun hanya 1 bit. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum.

6. *Distortion*

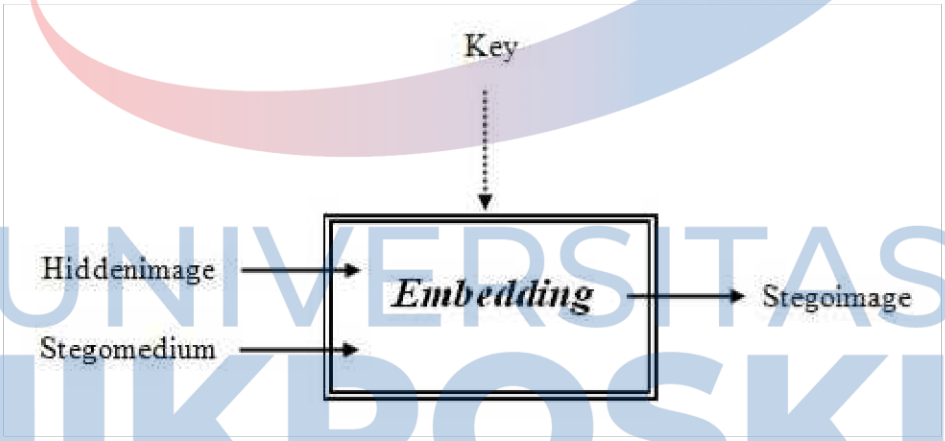
Metode ini menciptakan perubahan atas benda yang ditumpangioleh data rahasia.

7. *Cover Generation*

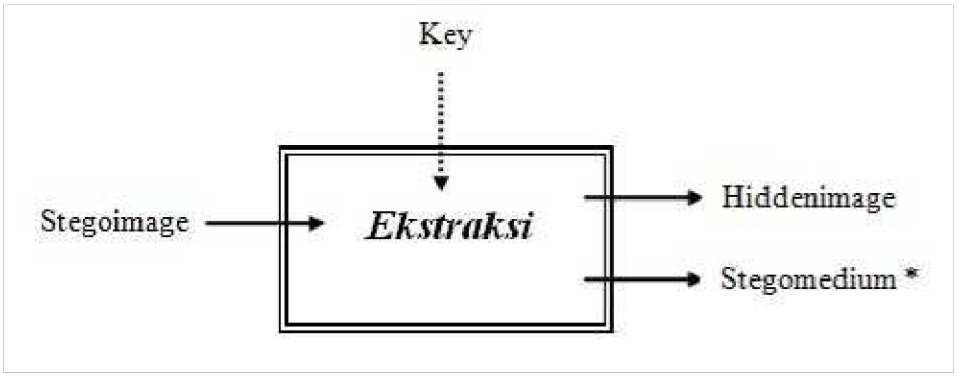
Metode ini lebih unik daripada metode lainnya karena *cover object* dipilih untuk menyembunyikan pesan. Contoh dari metode ini adalah *Spam Mimic*.

2.4 Proses Steganografi

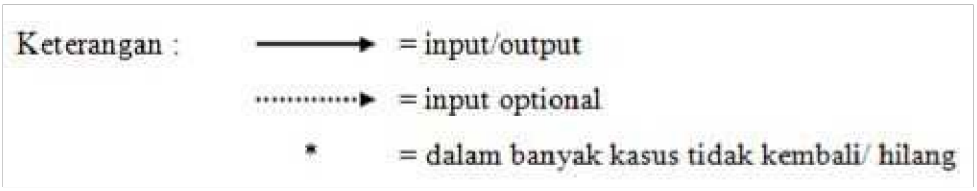
Secara umum, terdapat dua proses didalam *steganografi*, yaitu proses *embedding* untuk menyembunyikan pesan dan ekstraksi untuk mengekstraksi pesan yang disembunyikani. Proses-proses tersebut dapat dilihat pada gambar di bawah ini:



Gambar 2.2 Ilustrasi Proses Embedding.



Gambar 2.3 Ilustrasi Proses Ekstraksi



Gambar 2.4 Keterangan Ekstraksi.

Gambar 2.2 menunjukkan proses penyembunyian pesan dimana di bagian pertama, dilakukan proses *embedding hidden image* yang hendak disembunyikan secara rahasia ke dalam *stegomedium* sebagai media penyimpanan, dengan memasukkan kunci tertentu *(key)*, sehingga dihasilkan media dengan data tersembunyi di dalamnya (*stego image*). Pada Gambar 2.3, dilakukan proses ekstraksi pada *stego image* dengan memasukkan *key* yang sama sehingga didapatkan kembali *hidden image*. Kemudian dalam kebanyakan teknik steganografi, ekstraksi pesan tidak akan mengembalikan *stegomedium* awal persis sama dengan *stegomedium* setelah dilakukan ekstraksi bahkan sebagian besar mengalami kehilangan. Karena saat penyimpanan pesan tidak dilakukan pencatatan kondisi awal dari *stegomedium* yang digunakan untuk menyimpan pesan (Provos, 2003).

2.5 Metode Steganografi *Least Significant Bit*

Strategi penyembunyian data citra yang digunakan untuk menyisipkan citra kedalam media citra adalah dengan metode *Least Significant Bit*(LSB). Dimana bit data citra akan digantikandengan bit paling rendah dalam media citra. Pada file citra 24 bit setiap piksel pada citra terdiri dari susunan tiga warna, yaitu merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (1 byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Informasi dari warna biru berada pada bit 1 sampai bit 8, dan informasi warna hijau berada pada bit 9 sampai dengan bit 16, sedangkan informasi warna merah berada pada bit 17 sampai dengan bit 24. Menurut Kekre et al(2008) istilah algoritma substitusi LSB adalah skema yang paling sederhana untuk menyembunyikan pesan dalam sebuah citra host. Ia mengganti bit yang tidak signifikan dari masing-masing piksel dengan sedikit aliran pesan terenkripsi. Penerima dapat mengambil pesan dengan menguraikan LSB daris etiap piksel dari stego *image* dengan kunci yang diberikan. Karena hanya sedikit yang signifikan dari piksel yang berubah maka secara visual tidak terlihat oleh manusia.

Metode LSB merupakan teknik substitusi pada *steganografi*. Representasi warna dari piksel–piksel bisa diperoleh dari warna–warna primer, yaitu merah, hijau dan biru. Citra 24-bit menggunakan 3 byte untuk masing–masing piksel, dimana setiap warna primer direpresentasikan dengan ukuran 1 byte. Penggunaan citra 24-bit memungkinkan setiap piksel

direpresentasikan dengan nilai warna sebanyak 16.777.216. Dua bit dari saluran warna tersebut biasa digunakan menyembunyikan data yang akan mengubah jenis warna piksel-nya menjadi 64 warna. Hal itu akan mengakibatkan sedikit perbedaan yang tidak bisa dideteksi secara kasat mata oleh manusia (Ariyus, 2009).

Untuk menjelaskan metode ini,digunakan citra digital sebagai stegomedium. Pada setiap byte terdapat bit yang tidak signifikan. Misalnya pada byte 00011001, maka bit LSB-nya adalah 1. Untuk melakukan penyisipan pesan, bit yang paling tepat untuk diganti dengan bit pesan adalah bit LSB, sebab pengubahan bit tersebut hanya akan mengubah nilai byte-nya menjadi satu lebih tinggi atausatu lebih rendah. Sebagai contoh, urutan bit berikut ini menggambarkan 3 pikselpada stegomedium 24-bit.

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

Pesan yang akan disisipkan adalahkarakter A yang nilai biner-nya adalah 01000001 (ASCII), maka akan dihasilkan stegoimage dengan urutan bit sebagai berikut:

(00100110 11101001 11001000)

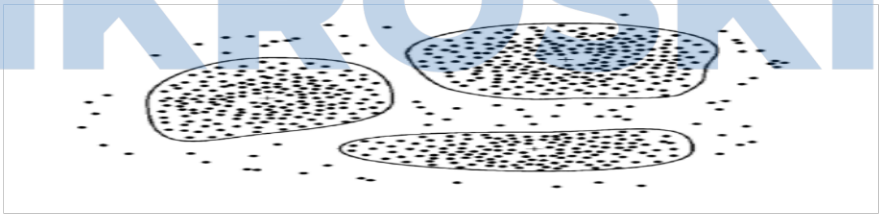
(00100110 11001000 11101000)

(11001000 00100111 11101001)

Terlihat hanya tiga bit rendah yang berubah (bit dengan garis bawah), untuk mata manusia maka tidak akan tampak perubahannya.

2.6 Pengertian Clustering

Clustering adalah proses pengelompokan satu set objek data ke dalam beberapa kelompok atau *cluster* sehingga objek dalam sebuah *cluster* memiliki



Gambar 2.5 Ilustrasi Proses Clustering.

jumlah kemiripan yang tinggi, tetapi sangat berbeda dengan objek di *cluster* lain. Ketidakmiripan dan kesamaan dinilai berdasarkan nilai atribut yang menggambarkan objek dan sering melibatkan perlakuan jarak. *Clustering* sebagaialat *data mining* dapat diterapkan pada berbagai bidang, seperti biologi, keamanan, intelijen bisnis, dan pencarian web (Han, J., 2012:443). Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas

dan meminimumkan kesamaan antar *cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruangmultidimensi. Beberapa algoritma pada teknik *clustering* menggunakan perhitungan jarak minimum untuk mengukur kemiripan antar data apabila data berupa data numeric (Kusnawi, 2007: 6). Menurut Jiawei Han (2006: 401-430) secara umum metode pada *clustering* dapat digolongkan ke dalam beberapa metode berikut:

1. Metode partisi (*Partitioning Method*)

Langkah kerja metode partisi, yaitu apabila terdapat basis data sejumlah *n* objek atau data tupelo, selanjutnya data di partisi menjadi *k* partisi dari data, dimana setiap partisi mewakili sebuah *cluster* dan $k \leq n$. Adapun syarat yang harus terpenuhi sebagai berikut: (1) setiap kelompok harus berisi setidaknya satu objek, dan (2) setiap objek harus memiliki tepat satu kelompok.Awalnya basis data dipartisi menjadi *k* partisi. Kemudian menggunakan teknik relokasi berulang, mencoba untuk memperbaiki partisi dengan memindahkan dari satu kelompok ke kelompok lain. Kriteria umum dari partisi yang baik adalah bahwa objek dalam satu *cluster* memiliki kemiripan yang sangat dekat, sedangkan objek dalam *cluster* yang berbeda memilikikemiripan yang jauh berbeda. Pencapaian optimalitas global dalam pengelompokan berbasis partisi akan memerlukan penghitungan lengkap dari semua partisi yang memungkinkan.Sebaliknya, sebagian besar aplikasi mengadopsi salah satu dari beberapa metode heuristik yang popular, seperti (1) algoritma *k-means,* dimana setiap segmen diwakili oleh nilai rata-rata dari objek dalam *cluster*, dan (2) algoritma *k-medoids*, dimana setiap segmen diwakili oleh salah satu objek yang terletak didekat *centroid*. Metode pengelompokan heuristik ini bekerja dengan baik untuk menemukan *cluster* berbentuk bola kecil untuk basis data yang berukuran sedang.

2. Metode Hirarki (*Hierarchi Method*)

Metode hirarki menciptakan dekomposisi hirarki dari himpunan objek data yang diberikan. Sebuah metode hirarki dapat diklasifikasikan sebagai salah satu *agglomerative* atau memecah belah, berdasarkan cara dekomposisi hirarki terbentuk. Pendekatan *agglomerative* memiliki dua cara pendekatan, yaitu *bottom–up* dan *top-down*. Pendekatan *bottom-up* berlangsung seperti berikut, awalnya setiap objek membentuk kelompok tersendiri. Berturut-turut menggabungkan objek atau kelompok yang dekat satu sama lain, sampai semua kelompok digabung menjadi satu (tingkat teratas dari hirarki), atau sampai terjadinya kondisi pemutusan hubungan. Sedangkan pendekatan *topdown*, dimulai dengan semua objek dalam *cluster* yang sama dibagi menjadi kelompok yang lebih kecil, sampai akhirnya setiap objek dalam satu *cluster* atau sampai terjadi kondisi pemutusan hubungan. Metode hirarki memuat fakta bahwa

setelah langkah penggabungan atau *split* dilakukan, proses memecah belah tidak dapat dibatalkan. Ada duapendekatan untuk meningkatkan kualitas pengelompokan hirarki: (1) melakukan analisis yang cermat terhadap objek "*linkage*" pada setiap partisi hirarki, seperti di Chameleon, atau (2) mengintegrasikan *aglomerasi* hirarki dan pendekatan-pendekatan lain dengan terlebih dahulu menggunakan algoritma *agglomerative* hirarki objek kelompok ke dalam *microclusters*, dan kemudian melakukan *macroclustering* pada *microclusters* menggunakan metode pengelompokan lain seperti relokasi berulang. Salah satu algoritma yang tergolong kedalam metode hirarki yaitu BIRCH (*Balanced Iterative Reducing and Clustering Using Hierarchies*). BIRCH merupakan salah satu algoritma pengelompokan hirarki yang terintegrasi. BIRCH memperkenalkan
dua konsep, *clustering feature* dan *clustering feature tree* (CF tree), yang mana digunakan untuk menggambarkan ringkasan *cluster*.

2.6.1 Pengertian K-Means *Clustering*

Pengelompokan objek (objek *clustering*) adalah salah satu proses dari objek *mining* yang bertujuan untuk mempartisi objek yang ada kedalam satu atau lebih *cluster* objek berdasarkan karakteristiknya. Objek dengan karakteristik yang sama dikelompokkan dalam satu *cluster* dan objek dengan karakteristik berbeda dikelompokkan kedalam *cluster* yang lain. *K-Means Clustering* termasuk dalam kelompok metode *cluster analysis non hirarki*, dimana jumlah kelompok yang akan dibentuk sudah terlebih dahulu diketahui atau ditetapkan jumlahnya. *K-Means Clustering* menggunakan metode perhitungan jarak (*distance*) untuk mengukur tingkat kedekatan antara objek dengan titik tengah (*centroid*). *K-Means* tidak terpengaruh terhadap urutan objek yang digunakan, hal ini dibuktikan ketika penulis mencoba menentukan secara acak titik awal pusat *cluster* dari salah satu objek pada permulaan perhitungan. Jumlah keanggotaan *cluster* yang dihasilkan berjumlah sama ketika menggunakan objek yang lain sebagai titik awal pusat cluster tersebut. Namun, hal ini hanya berpengaruh pada jumlah iterasi yang dilakukan. *K-Means Cluster* pada dasarnya dapat diterapkan pada permasalahan dalam memahami perilaku konsumen, mengidentifikasi peluang produk baru dipasaran dan *K-Means* ini juga dapat digunakan untuk meringkas objek dari jumlah besar sehingga lebih memudahkan untuk mendiskripsikan sifat-sifat atau karakteristik dari masing-masing kelompok.

2.6.2 Algoritma K Means *Clustering*

Pada algoritma ini, yang menjadi pusat cluster dinamakan centroid, centroid merupakan nilai acak dari seluruh kumpulan data yang dipilih pada tahap awal, kemudian *K-Means* menyeleksi masing-masing komponen dari seluruh data dan memisahkan data tersebut kedalam salah satu centroid yang sudah diuraikan sebelumnya berdasarkan jarak terdekat antara komponen data dan pusat masingmasing centroid dengan syarat tidak ada lagi data yang berpindah kelompok. Algoritma penglompokan data *K-means* adalah sebagai berikut :

1. Tentukan jumlah kelompok
2. Alokasikan data ke dalam kelompok secara acak
3. Hitung pusat cluster (centroid/rata-rata) dari data yang ada di masing-masing cluster
4. Alokasikan masing-masing data ke centroid/rata-rata terdekat
5. Kembali ke Langkah 3 :
   - apabila masih ada data yang berpindah cluster
   - atau apabila perubahan nilai centroid ada yang di atas nilai threshold yang ditentukan,
   - atau apabila perubahan nilai pada fungsi obyektif yang digunakan masih di atas nilai threshold yang ditentukan menggunakan rumus persamaan (2.4).

$$C_i = \frac{1}{M}\sum_{j=1}^{M} x_j$$

Pada langkah 3 dalam Algoritma di atas, lokasi centroid (titik pusat) setiap kelompok yang diambil dari rata-rata (*mean*) semua nilai data pada setiap fiturnya harus dihitung kembali. Jika M menyatakan jumlah data dalam sebuah cluster, i menyatakan fitur ke-i dalam sebuah cluster dan p menyatakan dimensi data, maka untuk menghitung centroid fitur ke-i digunakan persamaan 2.1.

$$J = \sum_{i=1}^{N}\sum_{l=1}^{K} a_{ic} D(x_i, C_l)^2$$

(2.1)

Formula tersebut dilakukan sebanyak p dimensi sehingga i mulai 1 sampai p. Cara mengukur jarak data ke pusat cluster menggunakan Euclidean (Bezdek, 1981) pada persamaan 2.2.

(2.2)

Keteranganpersamaan 2.2:

D = jarak antara data x2 dan x1, dan | . | adalah nilai mutlak.

P = Dimensi data

X2j = Koordinat dari obyek i pada dimensi k

X1j = Koordinat dari obyek j pada dimensi k

$$a_{il} = \begin{cases} 1 & d = \min\{D(x_i, C_l)\} \\ 0 & lainnya \end{cases}$$

Pada langkah 4 pada Algoritma, pengalokasian kembali data ke dalam masingmasing kelompok dalam metode K-means didasarkan pada perbandingan jarak antara data dengan sentroid setiap kelompok yang ada. Data dialokasikan ulang secara tegas ke kelompok yang mempunyai sentroid dengan jarak terdekat dari data tersebut. Pengalokasian data ke cluster menggunakan persamaan 2.3

$$D(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^{p} |x_{2j} - x_{1j}|^2} \quad (2.3)$$

Dimana a_{il} adalah nilai keanggotaan titik x_i ke pusat cluster C_i , d adalah jarak terpendek dari data x ¡ke K cluster setelah dibandingkan, dan Cl *centroid* (pusat cluster) ke-1.

Fungsi objektif berdasarkan jarak dan nilai keanggotaan data dalam cluster

(2.4)

Dimana N adalah jumlah data, K adalah jumlah *cluster*, a_{il} adalah nilai keanggotaan titik data x_i ke pusat cluster C_i, C_i adalah pusat cluster ke-*l*, $D(x_i , C_i)$ adalah jarak titik x_i ke cluster C_i yang diikuti. Untuk a mempunyai nilai 0 atau 1. Apabila suatu data merupakan anggota suatu kelompok maka nilai a_{i1} =1 , jika tidak, akan maka nilai a_{i1} = 0.

2.7 Kriptografi

Definisi kriptografi ada beberapa yang telah dikemukakan di dalam berbagai literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat, dan mata-mata. Namunsaat ini kriptografi lebih dari sekedar *privacy*, tetapi juga untuk tujuan data integrity, authentication, dan non-repudiation (Mollin ,2006). Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan, selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata seni di dalam definisi

tersebut maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata "graphy" di dalam "cryptography" itu sendiri sudah menyiratkan suatu seni (Munir, 2006).

Adapun istilah-istilah yang digunakan dalam kriptografi dalam melakukan proses kerjanya adalah sebagai berikut:

a. *Plaintext*

   Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

b. *Ciphertext*

   Ciphertext merupakan pesan yang telah disandikan (dikodekan) sehingga siap untuk dikirimkan.

c. Enkripsi

   Enkripsi merupakan proses yang dilakukan untuk menyandikan plaintext menjadi ciphertext dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

d. Dekripsi

   Dekripsi merupakan proses yang dilakukan untuk memperoleh kembali *plaintext* dari *ciphertext*.

e. Kunci

   Kunci yang dimaksud disini adalah kunci yang dipakai untuk melakukan dekripsi dan enkripsi. Kunci terbagi menjadi dua bagian, kunci pribadi (private key) dan kunci umum (*public key*).

f. Kriptosistem

   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

g. Kriptanalasis

   Kriptanalasis merupakan suatu ilmu untuk mendapatkan plainteks tanpaarus mengetahui kunci secara wajar (Munir, 2006).

2.7.1 Tujuan Kriptografi

Empat tujuan mendasar dari ilmu kriptografi yang juga merupakan aspek keamanan informasi, yaitu :

a. Kerahasiaan (*confidentiality*)

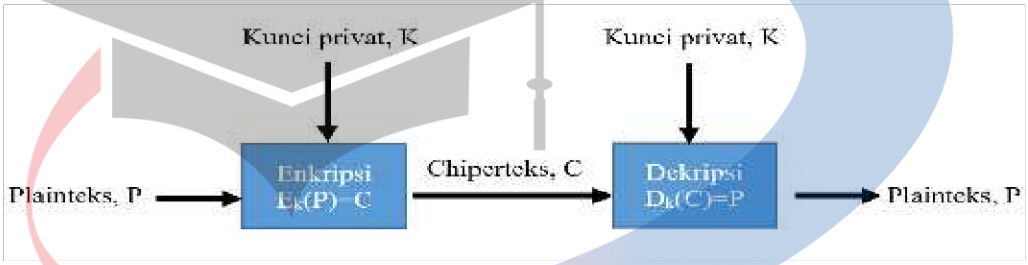
Kerahasiaan berarti data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja.

b. Otentikasi (*authentication*)

Pada saat mengirim atau menerima informasi, kedua belah pihak perlu mengetahui bahwa pengirim dari pesan tersebut adalah orang yang sebenarnya.

c. Integritas data (*integrity*)

Tuntutan integritas data ini berhubungan dengan jaminan setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebutyang diganti,



diduplikasi, dirusak, diubah urutannya dan ditambahkan.

d. Ketiadaan penyangkalan (*nonrepudiation*)

Nonrepudiation mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan/informasi (Munir,2006).

2.7.2 Algoritma Kriptografi

Algoritma kriptografi atau sering disebut dengan cipher adalah suatu fungsi matematis yang digunakan untuk melakukan enkripsi dan dekripsi (Stinson,1995). Algoritma kriptografi ada dua macam, yaitu algoritma simetris (*symmetric algorithms*) dan algoritma asimetris (*asymmetric algorithms*).

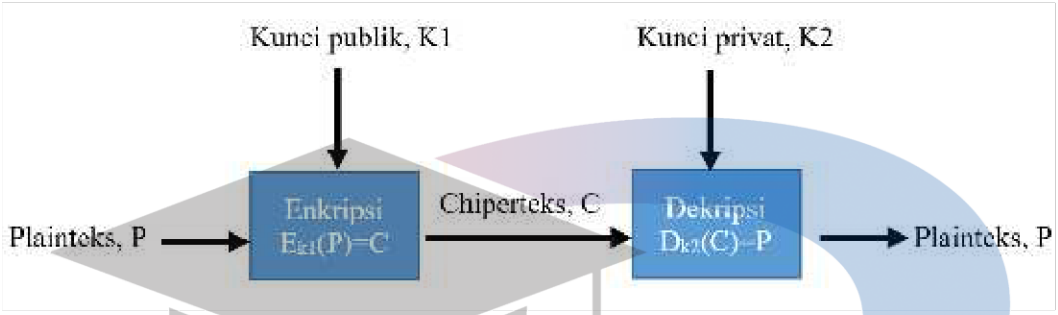
2.7.2.1 Algoritma Simetris

Algoritma simetris atau disebut juga algoritma konvensional adalah algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsi. Algoritma ini mengharuskan pengirim dan penerima menyetujui satu kunci tertentu sebelum dapat berkomunikasi secara aman. Keamanan algoritma simetri tergantung pada rahasia kunci. Pemecahan kunci berarti memungkinkan setiap orang dapat mengenkripsi dan mendekripsi pesan dengan mudah.

Gambar 2.6 Ilustrasi Algoritma Simestris.

2.7.2.2 Algoritma Asimetris

Algoritma Asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi dekripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia itu, yang dalam hal ini kunci rahasia, untuk melakukan pembongkaran terhadap kode yang dikirim untuknya.



Gambar 2.7 Ilustrasi Algoritma Asimetris.

2.8 Sistem Kriptografi

Menurut Stinson (1995), sistem kriptografi (*cryptosystem*) adalah suatu 5- tuple (*P, C, K, E, D*) yang memenuhi kondisi sebagai berikut :

1. *P* adalah himpunan *plaintext*,
2. *C* adalah himpunan *ciphertext*,
3. *K* atau ruang kunci (*keyspace*), adalah himpunan kunci,
4. *E* adalah himpunan fungsi enkripsi $ek : P \rightarrow C$,
5. *D* adalah himpunan fungsi dekripsi $dk : C \rightarrow P$,
6. Untuk setiap $k \in K$ terdapat $ek \in E$ dan $dk \in D$. Setiap $ek : P \rightarrow C$ dan $dk : C \rightarrow P$ merupakan fungsi sedemikian hingga $dk(ek(x)) = x$, untuk setiap *plaintext* $x \in P$.

Sistem kriptografi terdiri dari sebuah algoritma, seluruh kemungkinan *plaintext*,*ciphertext* dan kunci-kuncinya. Sistem kriptografi merupakan suatu fasilitas untuk mengkonversikan *plaintext* menjadi *ciphertext*, dan sebaliknya.

2.9 Algoritma AES

Algoritma AES merupakan algoritma chiper yang aman untuk melindungi data atau informasi yang bersifat rahasia. AES dipublikasikan oleh NIST (National Institute of Standard and Technology) pada tahun 2001 yang digunakan untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dibobol. Input dan output dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data dalam satu kelompok 128 bit tersebut disebut juga

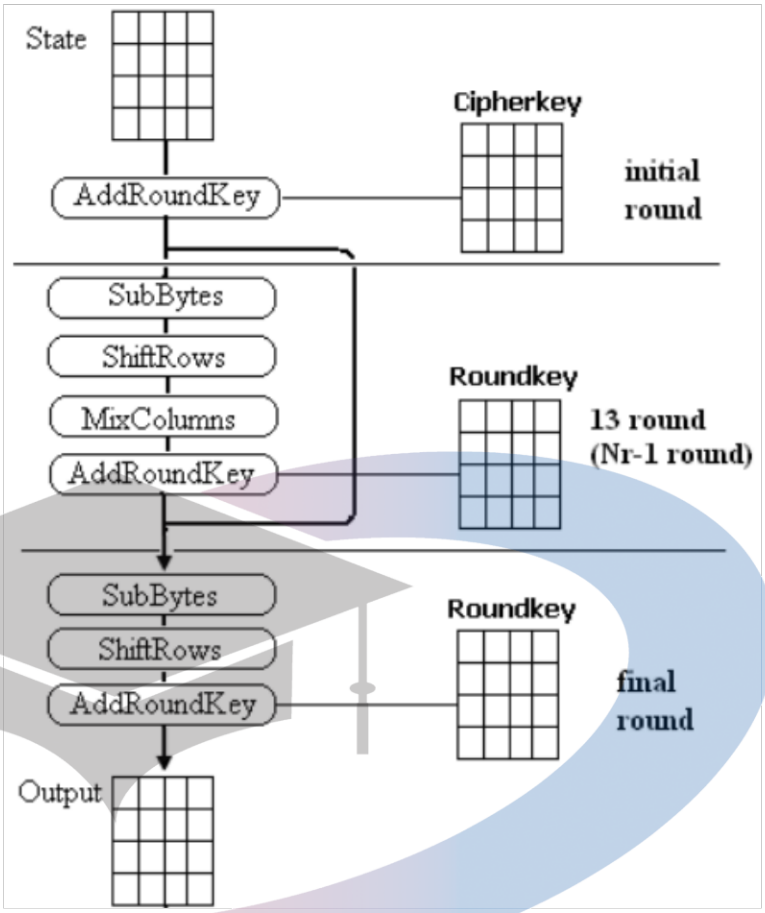
sebagai blok data atau plaintext yang nantinya akan dienkripsi menjadi chipertext. Panjang kunci dari AES terdiri dari panjang kunci 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini yang nantinya mempengaruhi jumlah putaran pada algoritma AES ini. Input dan output dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau plaintext yang nantinya akan dienkripsi menjadi chipertext. Panjang kunci dari AES terdiri dari panjang kunci 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini yang nantinya mempengaruhi jumlah putaran pada algoritma AES ini. Jumlah putaran yang digunakan algoritma ini ada tiga macam seperti pada tabel dibawah.

Tabel 2.1 Contoh Kunci AES, Panjang Kunci dan Jumlah Putaran

Tipe	Panjang Kunci	Panjang Blok Input	Jumlah Putaran
AES-128	128 bit	128 bit	10
AES-129	192 bit	128 bit	12
AES-256	256 bit	128 bit	14

2.9.1 Proses Enkripsi AES

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey*. Pada awal proses enkripsi, input yang telah dicopykan ke dalam state akan mengalami transformasi *byte AddRoundKey*. Setelah itu, state akan mengalami transformasi SubBytes, *ShiftRows, MixColumns, dan AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada round terakhir, state tidak mengalami transformasi *MixColumns*. (Yuniat, Indriyanta, & Rachmat, 2009).



Gambar 2.8 Proses Enkripsi AES.

a. *AddRoundKey*

Dalam initial round, transformasi *AddRoundKey*() dilakukan terhadap kunci utama. Sedangkan dalam 10 round yang lain, proses *AddRoundKey* dilakukan terhadap kunci putaran (round key). Proses *AddRoundKey* didefinisikan sebagai operasi XOR antara array state dengan round key. Operasi XOR dilakukan pada masingmasing *byte* dalam *array* sehingga menghasilkan nilai baru pada array hasil dengan ukuran array hasil sama dengan ukuran array state awal dan array key, yaitu sebesar 4x4. Hasil untuk masing-masing baris dan kolom pada *array* state hasil diperoleh dari hasil operasi XOR antara array state awal dengan *array* key untuk baris dan kolom yang sama

b. *SubBytes*

Transformasi *SubBytes*() memetakan setiap byte dari array state dengan menggunakan tabel substitusi S-Box. Tabel S-Box dapat dilihat pada berikut

Cara pensubstitusian adalah sebagai berikut:

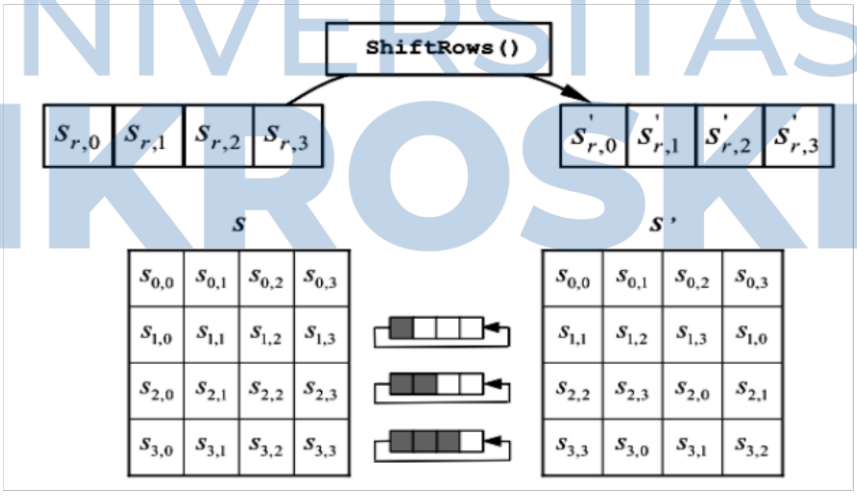
untuk setiap byte pada array, misalkan S[r,c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r,c], maka nilai substitusinya, yang dinyatakan dengan S'[r,c], adalah elemen di dalam S-Box yang merupakan perpotongan baris x dengan kolom y.

Tabel 2.2 S-Box.

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

c. ShiftRows

Transformasi *ShiftRows*() melakukan pergeseran secara wrapping (siklik) pada 3 baris terakhir dari array state. Jumlah pergeseran bergantung pada nilai baris (r). Baris r = 1 digeser sejauh 1 byte, baris r = 2 digeser sejauh 2 byte, dan baris r = 3 digeser sejauh 3 byte. Baris r = 0 tidak digeser.



Gambar 2.9 Transformasi ShiftRows.

d. MixColumn

Transformasi *MixColumns*() dilakukan setelah transformasi *ShiftRows*, merupakan sumber utama dari difusi pada algoritma AES. Difusi merupakan prinsip yang menyebarkan pengaruh satu bit plaintext atau kunci ke sebanyak mungkin *ciphertext*. *Transformasi MixColumns*() mengalikan setiap kolom dari array state dengan polinom a(x) mod (x4 + 1). Setiap kolom diperlakukan sebagai polinom 4 suku pada GF (28). Polinom a(x) yang ditetapkan pada persamaan berikut:

$$a(x) = \{03\}x3 + \{01\}\ x2 + \{01\}\ x + \{02\} \quad (1)$$

Transformasi ini dinyatakan sebagai perkalian matriks seperti pada persamaan berikut :

$$s'(x) = a(x) \otimes s(x) \quad (2)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Hasil dari perkalian matriks tersebut, setiap byte dalam kolom array state akan digantikan dengan nilai baru. Persamaan matematis untuk setiap byte tersebut pada persamaan berikut:

$$s'0,c = (\{02\} \bullet s0,c) \oplus (\{03\} \bullet s1,c) \oplus s2,c \oplus s3,c$$

$$s'1,c = s0,c \oplus (\{02\} \bullet s1,c) \oplus (\{03\} \bullet s2,c) \oplus s3,c$$

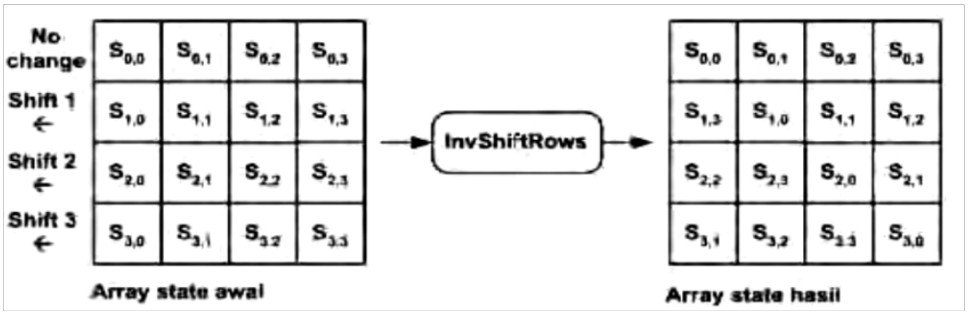
$$s'2,c = s0,c \oplus s1,c \oplus (\{02\} \bullet s2,c) \oplus (\{03\} \bullet s3,c)$$

$$s'3,c = (\{03\} \bullet s0,c) \oplus s1,c \oplus s2,c \oplus (*02\} \bullet s3,c) \quad (3)$$

Untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey. Algoritma dekripsi dapat dilihat pada skema berikut ini:

a. InvShiftRows

InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri. Ilustrasi transformasi InvShiftRows terdapat pada gambar berikut:



Gambar 2.10 Transformasi InvShiftRows.

b. InvSubBytes

InvSubBytes juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan tabel Inverse S-Box. Tabel Inverse S-Box akan ditunjukkan dalam tabel berikut:

Tabel 2.3 Inverse S-Box .

HEX		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	D5	30	36	A5	38	Bf	40	A3	9e	81	F3	D7	Fb
	1	7c	E3	39	82	9b	2f	Ff	87	34	8e	43	44	C4	De	E9	Cb
	2	54	7b	94	32	A6	C2	23	3d	Ee	4c	95	0b	42	Fa	C3	4e
	3	08	2e	A1	66	28	D9	24	B2	76	5b	A2	49	6d	8b	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5c	Cc	5d	65	B6	92
	5	6c	70	48	50	Fd	Ed	B9	Da	5e	15	46	57	A7	8d	9d	84
	6	90	D8	Ab	00	8c	Bc	D3	0a	F7	E4	58	05	B8	B3	45	06
	7	d0	2c	1e	8f	Ca	3f	0f	02	C1	Af	Bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	Dc	Ea	97	F2	Cf	Ce	F0	B4	E6	73
	9	96	Ac	74	22	E7	Ad	35	85	E2	F9	37	E8	1c	75	Df	6e
	a	47	F1	1a	71	1d	29	C5	89	6f	B7	62	0e	Aa	18	Be	1b
	b	fc	56	3e	4b	C6	D2	79	20	9a	Db	C0	Fe	78	Cd	5a	F4
	c	1f	Dd	A8	33	88	07	C7	31	B1	12	10	59	27	80	Ec	5f
	d	60	51	7f	A9	19	B5	4a	0d	2d	E5	7a	9f	93	C9	9c	Ef
	e	A0	E0	3b	4d	Ae	2a	F5	B0	C8	Eb	Bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	D6	26	E1	69	14	63	55	21	0c	7d

c. InvMixColumns

Setiap kolom dalam state dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan : Hasil dari perkalian matriks tersebut, setiap byte dalam kolom array state akan digantikan dengan nilai baru. Persamaan matematis untuk setiap byte tersebut pada persamaan berikut:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad (4)$$

s'0,c =({0E}• s0,c)⊕({0B}•s1,c)⊕({0D}•s2,c) ⊕({09}• s3,c)

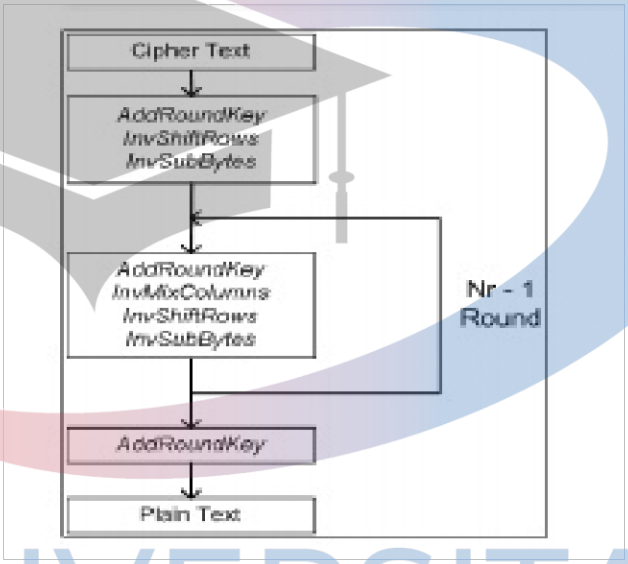
s'1,c =({09}•s0,c)⊕ ({0E} • s1,c) ⊕ ({0B} • s2,c) ⊕ ({0D}•s3,c)

s'2,c = ({0D}•s0,c)⊕ ({09}•s1,c)⊕ ({0E}• s2,c)⊕ ({0B}•s3,c)

s'3,c = ({0B}•s0,c) ⊕ ({0D}•s1,c)⊕ ({09}•s2,c) ⊕(*0E} • s3,c).

2.9.2 Proses Dekripsi AES

Transformasi cipher dapat dibalikkan dandiimplementasikan dalam arah yang berlawananuntuk menghasilkan inverse cipher yang mudahdipahami untuk algoritma AES. Transformasi byteyang digunakan pada invers cipher adalah*InvShiftRows*, InvSubBytes, InvMixColumns, danAddRoundKey. Algoritma dekripsi dapat dilihat padaskema berikut ini :

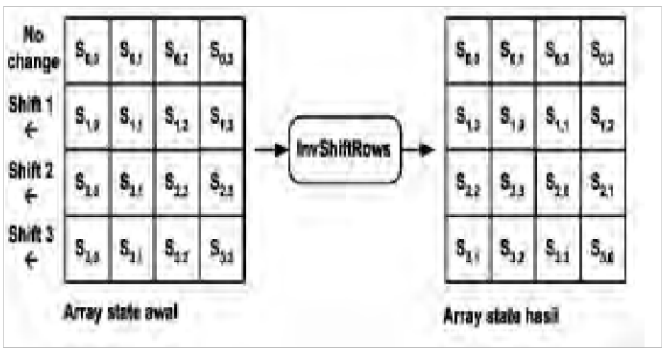


Gambar 2.11 Ilustrasi Proses Dekripsi.

a. *InvShifRows*

InvShiftRows adalah transformasi byte yangberkebalikan dengan transformasiShiftRows. Pada transformasi *InvShiftRows*,dilakukan pergeseran bit ke kanansedangkan pada ShiftRows dilakukanpergeseran bit ke kiri.

Ilustrasi transformasi *InvShiftRows* terdapat pada gambar berikut :



Gambar 2.12 Transformasi InvShifRows.

b. InvSubBytes

InvSubBytes juga merupakan transformasibytes yang berkebalikan dengantransformasi SubBytes. Pada *InvSubBytes*,tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse S-Box*. Tabel *Inverse S-Box* akan ditunjukkan dalam tabel berikut :

Tabel 2.4 Tabel Inverse S-Box

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

b. *InvMixColumns*

Setiap kolom dalam *state* dikalikan dengan matrik perkalian AES.:

Hasil dari perkalian dalam matrik adalah :

$$
\begin{aligned}
s'_{0,c} &= (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c}) \\
s'_{1,c} &= (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c}) \\
s'_{2,c} &= (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c}) \\
s'_{3,c} &= (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})
\end{aligned}
\tag{5}
$$

2.10 PSNR dan MSE

PSNR merupakan nilai perbandingan antara harga maksimum warna pada citra hasil filtering dengan kuantitas gangguan (*noise)* yang dinyatakan dalam satuan *decibel*(db), *noise* yang dimaksud adalah akar rata-rata kuadrat nilai kesalahan (*MSE*) (Santoso, I. 2013). PSNR secara umum digunakan sebagai ukuran dari kualitas dari rekonstruksi dalam pemrosesan citra. Nilai maksimum dari *pixel* dalam citra adalah 255. Semakin tinggi nilai PSNR akan menunjukkan bahwa kualitas citra hasil lebih baik dan hampir sama dengan citra aslinya. Kualitas citra dianggap rendah jika nilai PSNR-nya kurang dari 30dB dan kualitas citra dianggap tinggi jika nilai PSNR-nya di atas 40dB. Namun, jika nilai PSNR berada di antara

30dB sampai 40dB, maka kualitas citra masih dapat diterima (Cheddad et al, 2010). Untuk mendapatkan nilai PSNR dicari terlebih dahulu nilai *Mean Square Error* dari citra yang diuji. Hal ini lebih mudah didefinisikan dengan *Mean Square Error* (MSE), misal $I(x,y)$ adalah citra masukan dan $K(i,j)$ adalah citra keluaran, keduanya memiliki m baris dan n kolom, maka didefinisikan sebagai berikut :

$$MSE = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - I'(i,j)]^2 \quad (2.23)$$

Keterangan :

- m adalah jumlah baris dari citra I dan I'
- n adalah jumlah kolom dari citra I dan I'
- I adalah citra masukan
- I' adalah citra keluaran

Rumus untuk PSNR adalah :

$$PSNR = 20log_{10}\frac{255}{\sqrt{MSE}} \quad (2.24)$$

Keterangan :

MSE adalah *Mean Squared Error*, yang menunjukkan rata-rata *noise* yang terjadi antara citra I dan citra I'.