

BAB II

TINJAUAN PUSTAKA

2.1. Citra

Citra (*image*) adalah kumpulan dari titik yang mempunyai identitas tertentu untuk membentuk satu kesatuan perpaduan yang mempunyai arti, baik secara “artistik” maupun “intrinsik”. Citra yang baik adalah citra yang dapat menampilkan keindahan gambar (artistik) serta kejelasan gambar untuk penganalisaan dan maksud-maksud lainnya (intrinsik) (Munir, R., 2004).

Pixel menunjukkan satuan dari sebuah citra. *Pixel* merupakan singkatan dari *picture element*, yaitu unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per-inci. Setiap *pixel* mewakili tidak hanya satu titik dalam sebuah citra melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil yang biasa disebut dengan sel (Prasetyo, 2011).

Ada dua macam jenis citra yaitu citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu seperti hasil pengambilan gambar oleh kamera, citra tampilan di layar TV atau monitor (sinyal video). Sedangkan citra digital adalah citra yang disimpan di media penyimpanan dan dapat diolah computer (Prasetyo, 2011).

2.1.1. Pemrosesan Warna Citra

Penggunaan warna dalam pemrosesan citra dimotivasi oleh dua faktor utama. Pertama, warna adalah deskriptor yang *powerfull* yang sering menyederhanakan identifikasi objek dan pengekstrakan dari *scene*. Kedua, manusia dapat melihat dengan jelas ribuan bentuk warna dan intensitas, dibandingkan dengan hanya dua lusin bentuk grey. Faktor kedua ini penting dalam analisis citra (Prasetyo, 2011).

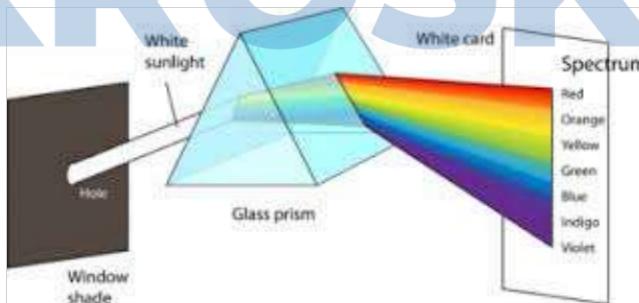
Pemrosesan citra berwarna dibagi dalam dua bagian penting: pemrosesan *full-color* dan *pseudo-color*. Kategori yang pertama, citra biasanya didapatkan dengan sensor *full-color*, seperti kamera TV atau *scanner* berwarna. Kategori yang

kedua, masalahnya satu pemberian warna ke intensitas monokrom tertentu atau range intensitas. Sampai saat ini, banyak pemrosesan warna citra digital dilakukan pada level *pseudocolor*. Bagaimanapun, pada dekade terakhir, sensor warna dan *hardware* untuk pemrosesan citra berwarna banyak tersedia dengan harga terjangkau. Hasilnya adalah bahwa teknik pemrosesan citra *full-color* sekarang digunakan dalam aplikasi secara luas, termasuk penerbitan, visualisasi, dan internet (Prasetyo, 2011).

2.1.1.1. Dasar-Dasar Warna

Walaupun proses diikuti oleh otak manusia dalam merasakan dan menginterpretasikan warna adalah fenomena *physiopsychological* yang belum sepenuhnya dimengerti, kealamian fisik warna dapat diekspresikan pada basis formal yang didukung oleh hasil eksperimen dan teori (Prasetyo, 2011).

Pada tahun 1966, Sir Isacc Newton mendiskusikan bahwa ketika cahaya dilewatkan ke kaca prisma, kemunculan cahaya tidak putih melainkan terdiri dari spektrum kontinu dari warna dengan range dari *violet* pada satu sisi sampai merah pada sisi lainnya. Sebagaimana yang ditunjukkan pada Gambar 2. 1, spektrum warna dapat dibagi menjadi tujuh luas daerah: *violet*, *indigo*, *blue*, *green*, *yellow*, *orange* dan *red*. Ketika ditampilkan dalam *full color* [Gambar 2. 1], tidak ada warna dalam akhir spektrum, melainkan setiap warna tercampur secara halus menjadi warna baru (Prasetyo, 2011).

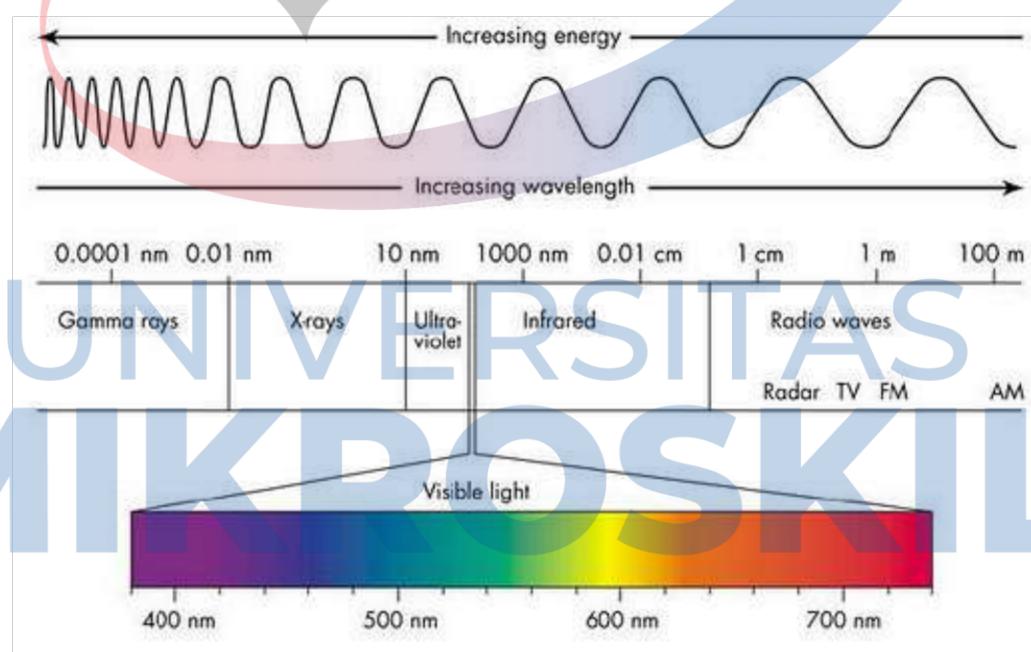


Gambar 2. 1 Spektrum warna yang dilewatkan prisma kaca

(Sumber: Prasetyo, 2011)

Dasarnya, warna yang dilihat manusia dan beberapa hewan dalam obyek ditentukan oleh keaslian cahaya yang dipantulkan dari obyek. Sebagaimana diilustrasikan pada Gambar 2. 2, cahaya yang tampak terdiri dari *band* yang relatif tipis frekuensinya dalam spektrum elektromagnetik. Benda yang memantulkan cahaya yang diseimbangkan dalam semua panjang gelombang *visible* menampilkan putih pada *observer*. Bagaimanapun, benda yang memantulkan, dibatasi range spektrum *visible* yang ditunjukkan beberapa bentuk warna. Misalnya, obyek *green* memantulkan cahaya dengan panjang gelombang utamanya dalam range 500 sampai 570 nm (Prasetyo, 2011).

Panjang gelombang yang dapat direspon oleh mata manusia berkisar dari 400 nm (biru) sampai 700 nm (merah) atau biasa disebut dengan istilah cahaya tampak (*visible spectrum*) seperti terlihat pada Gambar 2. 2 (Prasetyo, 2011).



Gambar 2. 2 Panjang gelombang *visible* spektrum elektromagnetik

(Sumber: Prasetyo, 2011)

2.1.1.2. Model Warna

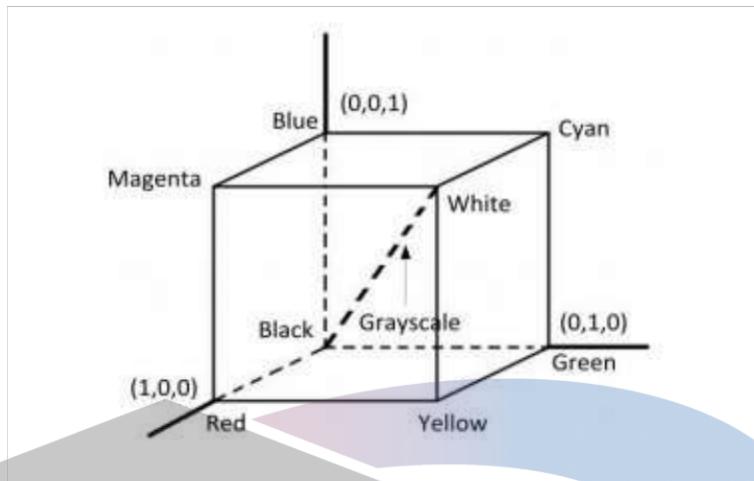
Tujuan dari model warna (disebut juga *color space* atau *color system*) adalah untuk memfasilitasi spesifikasi warna dalam beberapa standar. Esensinya,

model warna adalah representasi sistem koordinat dan *sub-space* di dalam sistem dimana setiap warna direpresentasikan oleh titik tunggal (Prasetyo, 2011).

Kebanyakan model warna yang digunakan saat ini diorientasikan pada *hardware* (seperti monitor berwarna dan printer) atau aplikasi dimana manipulasi warna adalah tujuannya. Terdapat banyak model warna diantaranya model RGB, CMY, HSI dan YIQ. Akan tetapi, dalam pengolahan citra digital, *hardware-oriented model* yang paling umum digunakan dalam prakteknya adalah model RGB (*Red, Green, Blue*) untuk monitor berwarna dan kamera video; model CMY (*Cyan, Magenta, Yellow*) dan CMYK (*Cyan, Magenta, Yellow, Black*) untuk pencerahan berwarna; dan model HSI (*Hue, Saturation, Intensity*) yang berhubungan dengan cara manusia menggambarkan dan menginterpretasikan warna dan mempunyai keuntungan bahwa model ini memecahkan informasi warna dan *grayscale* dalam citra. Adapun penjelasan dari ketiga model warna tersebut adalah sebagai berikut (Prasetyo, 2011):

a. Model Warna RGB

Dalam model RGB, setiap warna memperlihatkan komponen spektral *primary red, green* dan *blue*. Model ini didasarkan pada sistem koordinat kartesian. Sub-space warna yang dicari adalah kubus ditunjukkan pada Gambar 2. 3, dimana nilai RGB pada tiga sudut; *cyan, magenta* dan *yellow* ada pada tiga sudut lain; hitam adalah origin; dan putih adalah titik yang paling jauh dari origin. Dalam model ini, *grayscale* (titik-titik nilai equal RGB) diperluas dari hitam ke putih, sepanjang garis gabungan dua titik. Perbedaan warna dalam model ini adalah titik-titik yang berada di dalam kubus dan didefinisikan oleh penyebaran vektor dari origin. Untuk kemudahan, asumsinya adalah bahwa semua nilai warna dinormalisasi sehingga kubus yang ditampilkan pada Gambar 2. 3 adalah unit kubus. Maka, semua R, G dan B diasumsikan dalam range [0, 1] (Prasetyo, 2011).



Gambar 2. 3 Skema warna RGB dalam kubus

(Sumber: Prasetyo, 2011)

Citra yang direpresentasikan dalam model warna RGB terdiri dari tiga komponen citra, masing-masing untuk setiap warna primer (R, G, B). Ketika ditampilkan di monitor RGB, tiga kombinasi citra ini berada di layar fosfor untuk menghasilkan warna citra komposit. Jumlah bit yang digunakan untuk merepresentasikan setiap piksel dalam *space* RGB disebut *pixel depth*. Citra RGB dimana setiap citra red, green dan blue adalah citra 8-bit. Dalam kondisi setiap warna piksel RGB [maka, triplet dari nilai (R, G, B)] mempunyai kedalaman 24-bit (3 lapis citra dengan jumlah bit per lapis). Citra full-color sering digunakan untuk menyatakan citra berwarna RGB 24-bit (Prasetyo, 2011).

b. Model Warna CMY dan CMYK

Warna Cyan, magenta dan yellow adalah warna sekunder cahaya, atau merupakan warna primer dari pigmen. Misalnya ketika permukaan dilapisi dengan pigmen cyan diluminasi dengan cahaya putih, tidak ada cahaya merah yang dipantulkan dari permukaan. Hal tersebut karena cyan mengurangi cahaya merah dari cahaya putih yang dipantulkan, dimana dia terdiri dari perkiraan cahaya red, green dan blue (Prasetyo, 2011).

Banyak perangkat yang menyimpan pigmen berwarna pada kertas, seperti printer dan mesin fotokopi berwarna, membutuhkan data input CMY atau

melakukan konversi RGB ke CMY secara internal. Konversi ini dilakukan dengan operasi yang sederhana (Prasetyo, 2011):

Dimana asumsi bahwa semua nilai warna dinormalisasi dalam range [0, 1]. Persamaan di atas menunjukkan bahwa cahaya dipantulkan dari permukaan yang dilapisi dengan *cyan* murni tidak berisi *red* (karena $C = 1 - R$ dalam persamaannya). Dengan cara yang sama, magenta murni tidak memantulkan warna *green*, dan *yellow* murni tidak memantulkan warna *blue*. Persamaan di atas juga menyatakan bahwa nilai RGB bisa didapatkan dari satu set nilai CMY dengan mengurangkan setiap nilai dari nilai 1. Sebagaimana disebutkan sebelumnya, pengolahan citra dengan model warna ini digunakan dalam hubungan dengan pembuatan *output hardcopy*, sehingga operasi invers dari CMY ke RGB secara umum jarang digunakan dalam prakteknya (Prasetyo, 2011).

Jumlah yang sama untuk pigmen primer *cyan*, *magenta* dan *yellow* seharusnya menghasilkan hitam. Dalam prakteknya, kombinasi tiga warna ini untuk percetakan menghasilkan warna yang tidak sepenuhnya hitam sehingga dengan tujuan untuk menghasilkan warna hitam yang benar (yang biasanya lebih dominan dalam pencetakan), warna keempat, *black* ditambahkan, memberikan kemunculan model warna CMYK. Maka ketika publisher mengatakan “*four color printing*” maksudnya adalah tiga warna dari model warna CMY ditambah black (Prasetyo, 2011).

c. Model Warna HSI

Pembuatan warna dalam model RGB atau CMY dan perubahan dari satu model ke model yang lain adalah proses yang sederhana. Sistem warna ini secara ideal cocok untuk implementasi *hardware*. Sistem RGB cocok dengan kenyataan bahwa mata manusia dengan kuat dapat mengetahui warna primer *red*, *green* dan *blue*. Sayangnya, RGB, CMY dan model sejenis tidak secara baik cocok untuk menggambarkan warna dalam praktiknya untuk interpretasi manusia. Misalnya,

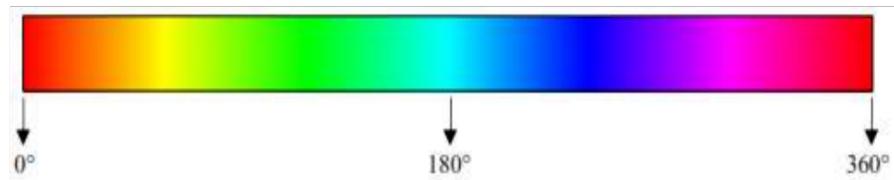
tidak mereferensikan ke warna dari *automobile* dengan pemberian persentase dari setiap warna primer yang mengisi warnanya. Selanjutnya, jangan dianggap bahwa citra berwarna terdiri dari tiga citra primer yang dikombinasikan untuk membentuk citra tunggal (Prasetyo, 2011).

Ketika manusia melihat objek berwarna, hal tersebut digambarkan oleh *hue*, *saturation* dan *brightness*. *Hue* adalah atribut warna yang menggambarkan warna murni (*yellow*, *orange* atau *red* murni), sedangkan *saturation* memberikan ukuran tingkat dimana warna murni yang dilemahkan oleh cahaya putih. *Intensity (gray-level)* adalah deskriptor yang paling berguna dalam citra *monochromatic*. Kuantitas ini secara definisi dapat diukur dan dengan mudah dapat diinterpretasikan. Model yang dibahas di atas disebut model warna HSI (*hue*, *saturation*, *intensity*), memisahkan komponen intensitas dari informasi warna yang dibawa (*hue* dan *saturasi*) dalam warna citra. Sebagai hasilnya, model HSI adalah *tool* yang ideal untuk mengembangkan algoritma pengolahan citra berdasarkan pada deskripsi warna yang alami dan intuitif terhadap manusia, pengembang dan *user*-nya. Adapun penjelasan dari ketiga komponen HSI (*hue*, *saturation*, *intensity*) adalah sebagai berikut (Hariyanto, 2009):

a. *Hue*

Menyatakan warna sebenarnya, seperti merah, violet, dan kuning. *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dan sebagainya dari cahaya. *Hue* berasosiasi dengan panjang gelombang cahaya, dan bila menyebut warna merah, violet, atau kuning, sebenarnya menspesifikasikan nilai *hue*-nya (Hariyanto, 2009).

Seperti terlihat pada Gambar 2. 4 di bawah ini, nilai *hue* merupakan sudut dari warna yang mempunyai rentang dari 0° sampai 360° . 0° menyatakan warna merah, lalu memutar nilai-nilai spektrum warna tersebut kembali lagi ke 0° untuk menyatakan merah lagi (Hariyanto, 2009).



Gambar 2. 4 Representasi nilai hue

(Sumber: Hariyanto, 2009)

b. *Saturation*

Menyatakan tingkat kemurnian warna cahaya, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Sebagai contoh, seperti terlihat pada Gambar 2. 5, warna merah adalah 100% warna-warna jenuh (*saturated color*), sedangkan warna *pink* adalah warna merah dengan tingkat kejemuhan sangat rendah (karena ada warna putih di dalamnya). Jadi, jika *hue* menyatakan warna sebenarnya, maka *saturation* menyatakan seberapa dalam warna tersebut (Hariyanto, 2009).

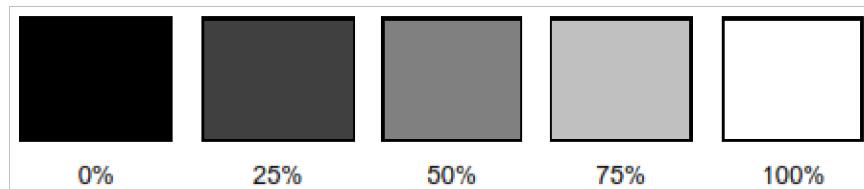


Gambar 2. 5 Representasi nilai saturation untuk warna merah

(Sumber: Hariyanto, 2009)

c. *Intensity/brightness/luminance*

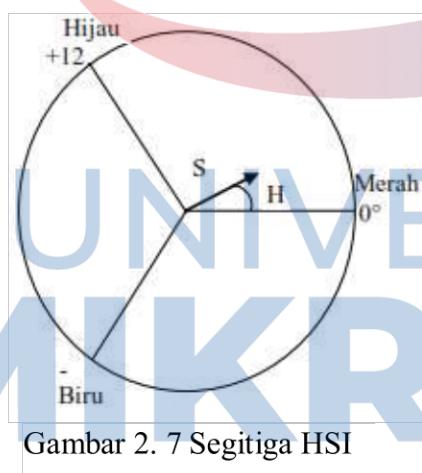
Atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa mempedulikan warna. Kisaran nilainya adalah antara gelap (hitam) dan terang (putih). Gambar 2. 6 memperlihatkan tingkatan nilai *Intensitas* dari 0% sampai dengan 100% (Hariyanto, 2009).



Gambar 2. 6 Representasi nilai intensity

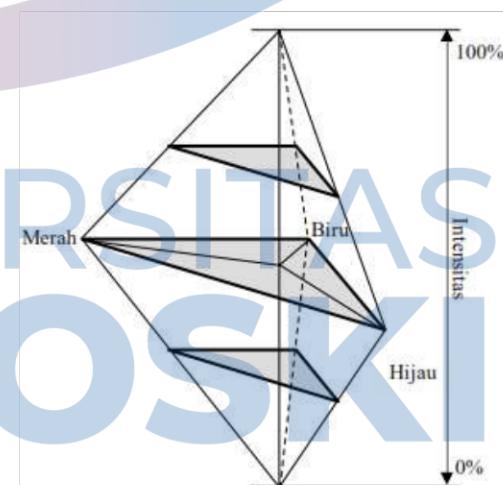
(Sumber: Hariyanto, 2009)

Intensitas adalah nilai abu-abu dari piksel dalam citra abu-abu. Segitiga HSI, seperti terlihat pada Gambar 2. 7 menampilkan kombinasi dari corak dan saturasi yang ditampilkan dari kombinasi warna pokok RGB. Sudut-sudut segitiga berkorespondensi dengan nilai maksimum dari warna-warna pokok (merah, hijau, biru) yang tersedia dalam komputer grafik. Piksel-piksel akromasitis (tidak mengandung warna) adalah bayangan abu-abu, berkorespondensi dengan jumlah ketiga warna pokok yang sama dan terletak di pusat segitiga (Hariyanto, 2009).



Gambar 2. 7 Segitiga HSI

(Sumber: Hariyanto, 2009)



Gambar 2. 8 Bentuk lengkap dari diagram HSI

(Sumber: Hariyanto, 2009)

Bentuk lengkap dari diagram HSI didapat dengan menambahkan dimensi *Intensitas* pada warna, dengan titik hitam pada dasarnya dan titik putih pada puncaknya. Bayangan abu-abu berada pada sepanjang sumbu vertikal, seperti diperlihatkan pada Gambar 2. 8. Diagram HSI menyempit menjadi sebuah titik pada

dasarnya dan juga pada puncaknya sebab hitam dan putih hanya dapat ditampilkan dengan kombinasi warna pokok RGB yang unik, yaitu ketiganya harus sama kuat dan berada pada nilai-nilai ekstrim. Bila Intensitas ketiga warna pokok bernilai penuh (posisi pada puncak sumbu vertikal), maka akan ditampilkan warna putih. Sedangkan bila Intensitas ketiga warna pokok bernilai minimum (posisi pada dasar sumbu vertikal), maka akan ditampilkan warna hitam. Apabila Intensitas ketiga warna sama kuat, tetapi tidak berada pada kedua titik ekstrim tadi, maka warna yang muncul adalah abu-abu. Pada dua keadaan ekstrim tersebut, dan juga banyak keadaan diantaranya, warna lain tidak muncul sebab kekuatan ketiga warna pokok selalu berimbang, dan kekuatan yang berimbang dari ketiga warna pokok meniadakan eksistensi warna penyusunnya karena tidak ada satu pun yang mendominasi (Hariyanto, 2009).

2.1.1.3. Konversi Model Warna RGB ke HSI

Komponen warna RGB pada citra dapat dikonversi menjadi model warna HSI. Untuk mendapatkan nilai Hue, Saturasi dan Intensitas, besarnya dapat dihitung secara langsung dengan (Muhammad et al., 2015):

UNIVERSITAS MIKROSKIL

$$h = \cos^{-1} \left[\frac{0.5 x \{(r-g)+(r-b)\}}{\sqrt{[(r-g)^2 + (r-b)(g-b)]^2}} \right], \text{ where } h \in [0, \pi] \text{ for } b \leq g$$

$$h = 2\pi - \cos^{-1} \left[\frac{0.5 x \{(r-g)+(r-b)\}}{\sqrt{[(r-g)^2 + (r-b)(g-b)]^2}} \right], \quad h \in [0, 2\pi] \text{ for } b > g$$

$$s = 1 - 3 \times \text{Min}(r, g, b), s \in [0, 1]$$

2.1.1.4. Konversi Model Warna HSI ke RGB

Sementara komponen warna HSI dapat dikonversi menjadi model warna RGB. Untuk mendapatkan nilai R, G dan B, besarnya dapat dihitung secara langsung dengan (Muhammad et al., 2015):

$$o = i \times (1 - s)$$

$$p = i \times \left[1 + \frac{s \times \cos(h)}{\cos\left(\frac{\pi}{3}h\right)} \right]$$

If $h < \frac{2\pi}{3}$ maka

If $\frac{2\pi}{3} \leq h < \frac{4\pi}{3}$ maka

$$h = h - \frac{2\pi}{3}$$

$r = 0$

$$g = p$$

If $\frac{4\pi}{3} < h \leq 2\pi$ maka

$$h = h - 240$$

$$g = 0$$

$$b = p$$

2.1.2. Noise Pada Citra

Noise sesungguhnya adalah komponen di citra yang tidak dikehendaki. Dalam praktik, kehadiran *noise* tidak dapat dihindari. Sumber *noise* pada citra digital bisa terjadi sejak pengambilan dan atau transmisi citra. Kinerja dari sensor citra dipengaruhi oleh banyak faktor seperti kondisi lingkungan selama pengambilan citra dan oleh kualitas sensitivitas elemen itu sendiri. Sebagai contoh, dalam pengambilan citra dengan kamera CCD, level pencahayaan dan suhu sensor adalah faktor utama yang mempengaruhi tingkat *noise* pada citra yang dihasilkan. Citra yang terkorupsi selama transmisi secara prinsip disebabkan interferensi channel yang digunakan untuk transmisi. Misalnya, citra yang ditransmisikan menggunakan jaringan wireless dapat terkorupsi sebagai hasil dari pencahayaan atau pengaruh atmosfer yang lain (Listiyani, 2007).

Berdasarkan bentuk dan karakteristiknya, *noise* pada citra dibedakan menjadi beberapa macam yaitu *Uniform*, *Gaussian*, *Speckle*, dan *Salt & Pepper* (Listiyani, 2007).

2.1.2.1. Noise Uniform

Noise Uniform dibangkitkan dengan cara membangkitkan bilangan acak $[0, 1]$ dengan distribusi *uniform*. Kemudian untuk titik-titik yang terkena *noise*, nilai fungsi citra ditambahkan dengan nilai *noise* yang ada, atau dirumuskan dengan (Listiyani, 2007):

Dimana:

- a : nilai bilangan acak berdistribusi *uniform* dari *noise*
 - p : prosentase *noise*
 - $y(i,j)$: nilai citra terkena *noise*
 - $x(i,j)$: nilai citra sebelum terkena *noise*

Hasil dari prosentase *noise uniform* untuk gambar kucing dapat dilihat pada gambar 2. 9 berikut ini:



Gambar 2. 9 Contoh Noise Uniform Dengan Presentase 10% Dan 50%.

(Sumber: lecturer.eepis-its.edu)

2.1.2.2. Noise Gaussian

Noise Gaussian merupakan model *noise* yang mengikuti distribusi normal standar dengan rata-rata nol dan standar deviasi 1. Efek dari *noise* ini adalah munculnya titik-titik berwarna yang jumlahnya sama dengan persentase pada *noise*. *Noise Gaussian* dibangkitkan dengan cara membangkitkan bilangan acak [0, 1] dengan distribusi *gaussian*. Kemudian untuk titik-titik yang terkena *noise*, nilai fungsi citra ditambahkan dengan nilai *noise* yang ada, atau dirumuskan dengan (Listiyani, 2007):

Dimana:

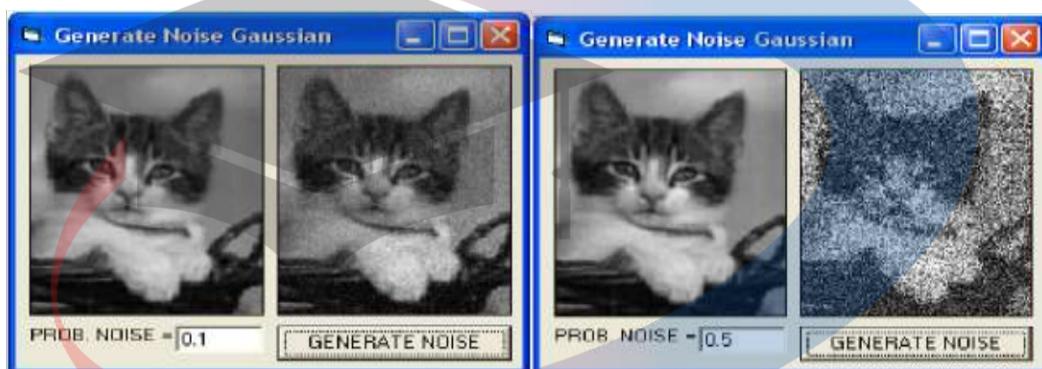
a : nilai bilangan acak berdistribusi *Gaussian*

p : prosentase noise

$y(i,j)$: nilai citra terkena *noise*

$x(i, j)$: nilai citra sebelum terkena *noise*

Hasil dari prosentase *noise gaussian* untuk gambar kucing dapat dilihat pada gambar 2. 10 berikut ini:



Gambar 2. 10 Contoh Noise Gaussian Dengan Prosentase 10% Dan 50%.

(Sumber: lecturer.eepis-its.edu)

2.1.2.3. Noise Speckle

Noise Speckle merupakan model *noise* yang memberikan warna hitam pada titik yang terkena *noise*. *Noise Speckle* dapat dibangkitkan dengan cara membangkitkan bilangan 0 (warna hitam) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas *noise*, dan dirumuskan dengan (Listiyani, 2007):

Dimana:

$F(x, y)$: nilai gray-scale pada titik (x, y)

$p(x, y)$: probabilitas acak

Hasil dari prosentase *noise speckle* untuk gambar kucing dapat dilihat pada gambar 2. 11 berikut ini:



Gambar 2. 11 Contoh Noise Speckle Dengan Prosentase 10% Dan 50%.

(Sumber: lecturer.eepis-its.edu)

2.1.2.4. Noise Salt & Pepper

Noise Salt & Pepper seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena *noise*. *Noise Salt & Pepper* dapat dibangkitkan dengan cara membangkitkan bilangan 255 (warna putih) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas *noise*, dan dirumuskan dengan (Listiyani, 2007):

Dimana:

$F(x, y)$: nilai gray-scale pada titik (x, y)

$p(x, y)$: probabilitas acak

Hasil dari prosentase *noise salt & pepper* untuk gambar kucing dapat dilihat pada gambar 2. 12 berikut ini:



Gambar 2. 12 Contoh *Noise Salt & Pepper* Dengan Prosentase 10% Dan 50%.

(Sumber: lecturer.eepis-its.edu)

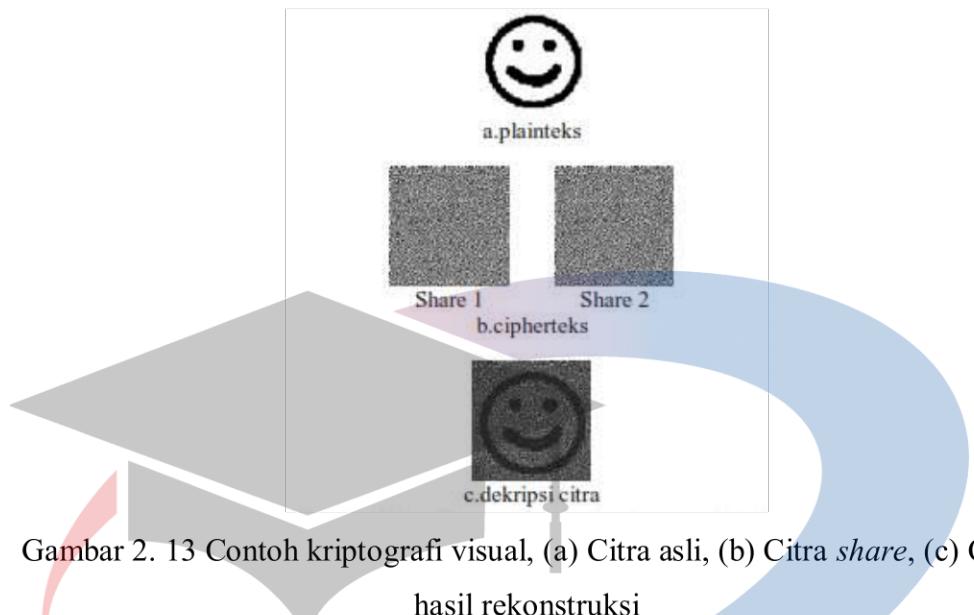
2.2. Kriptografi Visual

Kriptografi visual adalah skema pembagian yang digunakan dalam distribusi gambar. Kriptografi visual merupakan salah satu perluasan dari *secret sharing* yang diimplementasikan untuk suatu citra. Seperti halnya teknik kriptografi yang lain, kriptografi visual memiliki persyaratan kerahasiaan, integritas data dan otentifikasi. Kriptografi visual yaitu teknik kriptografi data berupa gambar atau citra dengan membagi gambar tersebut menjadi beberapa bagian. Setiap bagian gambar tersebut merupakan *subset* dari gambar aslinya (Hakim, 2014).

Kriptografi visual diperkenalkan pertama kali oleh Moni Naor dan Adi Shamir dalam paper mereka yang berjudul *Visual Cryptography*, dimuat dalam jurnal *Eurocrypt'94*, pada tahun 1995. Berbeda dengan kebanyakan algoritma kriptografi, algoritma ini tidak membutuhkan perhitungan rumit untuk mendekripsi pesan, tetapi hanya menggunakan sistem penglihatan manusia (Hakim, 2014).

Versi dasar kriptografi visual mempresentasikan *secret sharing* (2, 2). Maksudnya skema tersebut menghasilkan 2 (dua) citra pembagi dari gambar aslinya (P) yaitu sebuah gambar hitam putih. Dimana gambar P1 untuk bagian gambar 1 dan P2 untuk bagian gambar 2. P1 dan P2 merupakan distribusi acak dari piksel hitam putih dan tidak menunjukkan informasi apapun. Namun saat P1 dan P2 dilapiskan/ ditumpuk, maka akan didapat informasi seperti gambar aslinya. Apabila hanya ada P1, maka informasi P tidak dapat diketahui tanpa ada P2. Berikut adalah

ilustrasi citra asal (*plaintext*), citra tersandikan (*ciphertext*) dan citra dekripsi (Hakim, 2014):



2.2.1. Proses Pembentukan Share

Ada beberapa model teknik kriptografi visual yang diperkenalkan. Cara yang pertama adalah model sederhana. Metode ini menggunakan citra biner, yakni hanya memiliki warna hitam dan putih. Pembangkitan *share* dilakukan dengan bantuan representasi citra ke matriks secara acak. Representasi hitam dengan menggunakan angka 1 dan putih dengan menggunakan angka 0. Setiap piksel pada *secret image* akan diperlakukan secara terpisah. Masing-masing piksel tersebut akan muncul dalam n buah variasi, dinamakan *share*. Setiap *share* memiliki m buah *subpixel* berwarna hitam dan putih yang dicetak secara berdekatan sehingga sistem penglihatan manusia akan memandang rata distribusi warna hitam dan putih tersebut. Hasilnya dapat dimodelkan dalam matriks Boolean S berukuran $n \times m$, di mana $S[i, j] = 1$, jika dan hanya jika *subpixel* ke- j , pada *share* ke- i berwarna hitam. Jumlah baris pada matriks tersebut menyatakan banyaknya *share* yang dihasilkan dan jumlah kolom menyatakan jumlah *subpixel* pada masing-masing *share* (Romdhoni, 2008).

Dari model tersebut, penumpukan *share* dapat dianggap sebagai hasil fungsi OR pada baris-baris terkait dari matriks S tersebut. Hal ini sesuai bahwa warna hitam pada satu *subpixel* tidak dapat dihilangkan oleh warna putih pada *subpixel* lain yang bertumpuk dengannya. Tingkat keabu-abuan yang dihasilkan dari penumpukan ini akan dianggap sebagai warna hitam jika memenuhi bobot $H(V) \geq d$ dan dianggap warna putih jika memenuhi bobot $H(V) < d-am$ ($1 \leq d \leq m$) (Romdhoni, 2008). Hal ini dapat digambarkan sebagaimana dalam Gambar 2. 14 berikut:

Pixel	Probability	Share ₁	Share ₂	Share ₁ ⊕ Share ₂
	50%			
	50%			
	50%			
	50%			

Gambar 2. 14 Model kriptografi visual

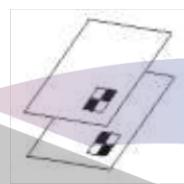
(Sumber: Romdhoni, 2008)

Model ini sangat sederhana sehingga tidak aman. Oleh karena itu diciptakan lagi teknik yang baru oleh Shamir dan Noor. Cara yang kedua yang merupakan perbaikannya adalah teknik kriptografi visual dengan menggunakan metode ekspansi piksel. Pada metode yang digunakan ini akan dihasilkan *noise* yang membuat hasil dekripsi citra tidak sama persis dengan plainteksnya karena adanya piksel tambahan hasil enkripsi. Pada metode ini satu piksel dibagi menjadi beberapa sub piksel tergantung banyak *share*-nya sebagai pengganti matriks yang sebelumnya diajukan. Gambar 2. 15 merupakan contoh satu piksel yang diekspan menjadi 4 subpiksel (Winata, 2012).



Gambar 2. 15 Pembentukan subpiksel dengan $m=4$

(Sumber: Winata, 2012)



Gambar 2. 16 Memecah warna hitam menjadi beberapa bagian

(Sumber: Winata, 2012)



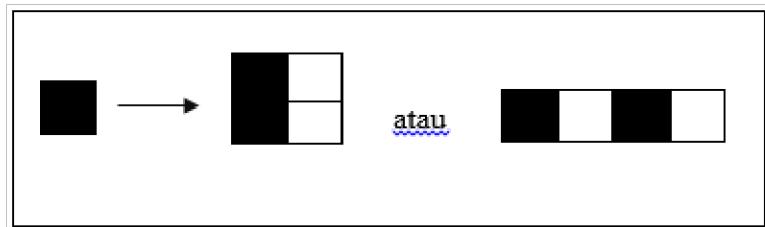
Gambar 2. 17 Dekripsi piksel

(Sumber: Winata, 2012)

UNIVERSITAS MIKROSKIL

Dengan melakukan ekspansi piksel, setiap satu piksel pada citra digital

diperluas menjadi satu blok piksel yang terdiri dari m piksel. Masing-masing piksel pada blok piksel kemudian disebut sebagai *subpixel*. Ekspansi piksel mengakibatkan citra baru hasil ekspansi mengalami perbesaran sebesar m -kali dari citra awal. Untuk contoh ekspansi piksel dengan $m = 4$, setiap satu piksel di citra awal diekspansikan menjadi empat piksel di citra baru. Hasil ekspansi setiap satu piksel tersebut dapat disusun dalam bentuk 2×2 piksel atau 1×4 piksel. Gambar 2. 18 menunjukkan contoh ekspansi piksel hitam dengan $m = 4$ (Sutanto, 2012).



Gambar 2. 18 Contoh Ekspansi piksel hitam ke 4 subpixels

(Sumber: Sutanto, 2012).

Jika piksel hitam direpresentasikan oleh angka 1 dan piksel putih direpresentasikan dengan angka 0, maka gambar di atas dapat direpresentasikan dalam matriks berikut (Sutanto, 2012):

$$[1] \longrightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \text{ atau } [1 \ 0 \ 1 \ 0]$$

Teknik kriptografi ini memiliki beberapa kelemahan. Salah satunya adalah *noise*. Seperti yang telah disebutkan sebelumnya, *noise* ini mengakibatkan hasil dekripsi citra tidak sesuai dengan plainteks yang didekripsi. Hal ini terjadi pada metode ekspansi. Kelemaan kedua yang ada dalam kriptografi visual ini adalah bahwa *share* yang dihasilkan tidak memiliki makna. Dengan begitu, tentu akan menimbulkan kecurigaan bagi orang lain yang mendapatkan *share* ini dan berasumsi terdapat pesan rahasia dalam *share* yang ia dapat (Winata, 2012).

2.2.2. Skema kriptografi visual (k, n)

Skema kriptografi visual (k, n) adalah sebuah teknik kriptografi dimana citra digital dibagi ke dalam n *shares* yang dilakukan dengan komputasi kriptografi dimana untuk mendekripsi citra diperlukan minimal k buah citra hasil tersebut. Jika terdapat q citra hasil, dimana $q < k$, maka tidak ada informasi apapun yang dapat diperoleh tentang citra asli (Kandar dan Maiti, 2011).

Solusi untuk skema k dari n terdiri dari dua buah matriks *Boolean* yang berukuran $n \times m$, $C0$ dan $C1$. Untuk membagi *pixel* putih, dipilih salah satu dari matriks $C0$, dan untuk membagi *pixel* hitam dipilih salah satu dari matriks $C1$.

Matriks yang telah dipilih merupakan representasi dari m *subpixel* yang terletak pada masing-masing n transparansi. Solusi benar apabila memenuhi seluruh kondisi berikut (Naor dan Shamir, 1995) :

1. Untuk sembarang matriks m pada $C0$, hasil operasi “or” dengan V pada sembarang baris k dari n memenuhi $H(V) < d - \alpha m$.
2. Untuk sembarang matriks m pada $C1$, hasil operasi “or” dengan V pada sembarang baris k dari n memenuhi $H(V) < d - \alpha m$.
3. Untuk sembarang $j < k$ baris yang dipilih, submatriksnya muncul dengan frekuensi yang sama pada $C0$ dan $C1$.

2.3. Skema kriptografi visual $((n-1, 1), n)$

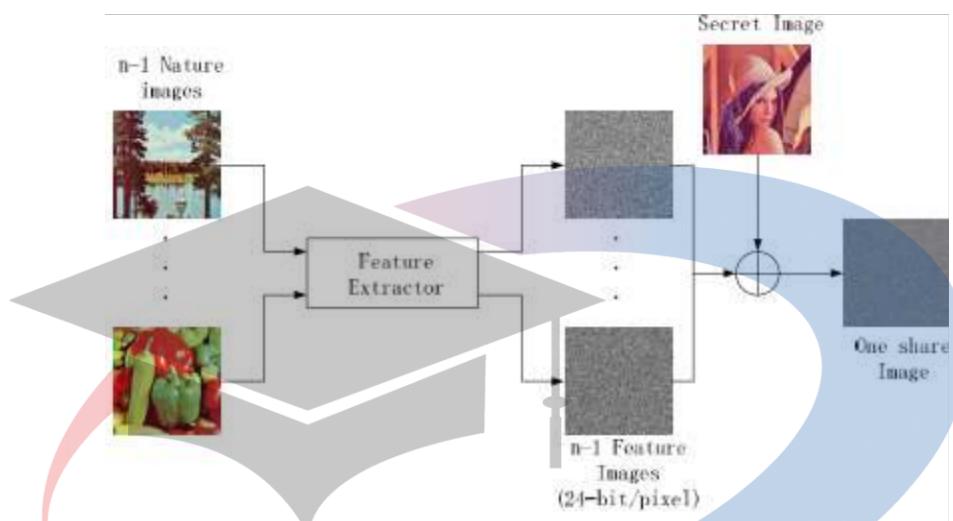
Skema kriptografi visual $((n-1, 1), n)$ merupakan skema kriptografi dimana $(n-1)$ *natural image* digunakan sebagai *input* untuk menghasilkan n buah *shares*, angka 1 menunjukkan jumlah *secret image*, sementara n menunjukkan jumlah *shares* yang dihasilkan yaitu *natural share image* dan *share image* hasil pembentukan *share*. Proses pembentukan *share* dilakukan dengan melakukan ekstraksi fitur untuk masing-masing dari $(n-1)$ *natural image* sehingga menghasilkan $(n-1)$ *natural share image*, kemudian dilakukan penggabungan. Hasil penggabungan kemudian dilakukan operasi XOR dengan *secret image* sehingga menghasilkan *share image* (Liu et al., 2013).

Sedangkan untuk proses rekonstruksi, $(n-1)$ *natural share image* digabungkan kemudian dilakukan operasi XOR dengan *share image* sehingga menghasilkan *secret image*. *Natural share image* yang berbeda atau kurang dari $(n-1)$ tidak akan mendapatkan informasi apapun tentang *secret image* (Liu et al., 2013).

2.3.1. Proses Pembentukan *Share*

Proses pembentukan *share image* dijelaskan pada Gambar 2. 19 yang terdiri dari dua tahapan, yaitu Natural Image Feature Extraction dan Encryption. Pada proses Natural Image Feature Extraction, masing-masing dari $(n-1)$ *natural image* dilakukan ekstraksi fitur sehingga menghasilkan $(n-1)$ *natural share*

image kemudian dilakukan penggabungan. Pada proses Encryption, hasil penggabungan *natural share image* dilakukan operasi XOR dengan *secret image* sehingga menghasilkan *share image* (Liu et al., 2013).

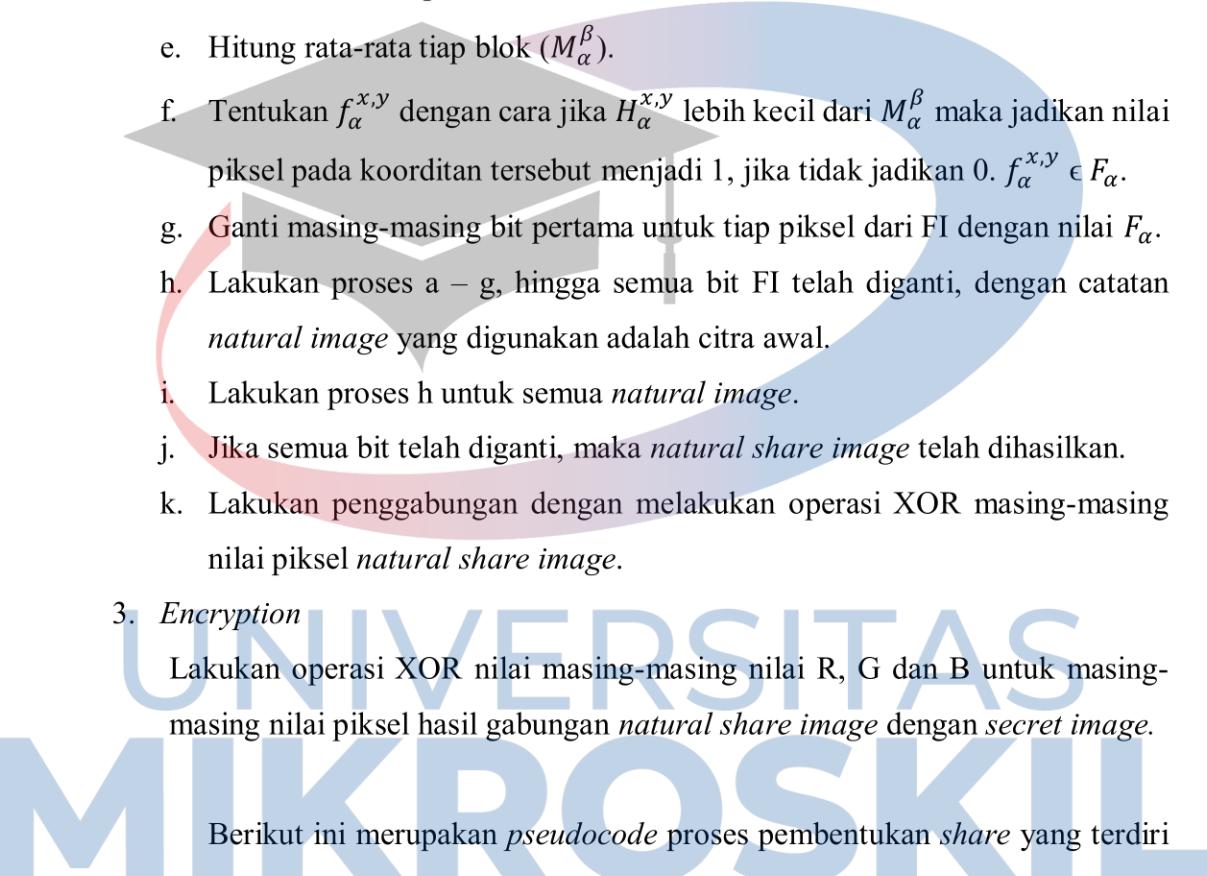


Gambar 2. 19 Proses Pembentukan *Share*

(Sumber: Liu et al., 2013)

Adapun tahapan proses pembentukan *share* dijelaskan sebagai berikut (Liu et al., 2013):

1. Inisialisasi nilai FI (*Feature Image*) dengan nilai 0 untuk masing-masing nilai R, G dan B. FI merupakan pemetaan dari *natural image*, jadi ukuran dan jumlah FI sama dengan ukuran dan jumlah *natural image*. Hasil akhir dari FI nantinya dijadikan sebagai *natural share image*.
2. *Natural Image Feature Extraction*
 - a. Lakukan pengacakan nilai piksel untuk masing-masing nilai R, G dan B pada masing-masing *natural image*. Pengacakan dilakukan dengan mengambil secara acak dua koordinat, lalu menukar kedua nilai piksel pada koordinat tersebut. Hal ini dilakukan sampai semua piksel telah tertukar, dengan catatan tidak ada nilai piksel yang ditukar lebih dari satu kali.

- 
- b. Lakukan operasi XOR untuk masing-masing nilai R, G dan B, dimana R di XOR dengan R sebelumnya, begitu seterusnya hingga semua piksel selesai dibaca, begitu juga dengan nilai G dan B.
 - c. Hitung $H_{\alpha}^{x,y}$ merupakan penjumlahan nilai R, G dan B pada suatu koordinat (x, y).
 - d. Bagi *natural image* menjadi beberapa blok, dimana masing-masing blok terdiri dari 8×8 piksel.
 - e. Hitung rata-rata tiap blok (M_{α}^{β}).
 - f. Tentukan $f_{\alpha}^{x,y}$ dengan cara jika $H_{\alpha}^{x,y}$ lebih kecil dari M_{α}^{β} maka jadikan nilai piksel pada koordinat tersebut menjadi 1, jika tidak jadikan 0. $f_{\alpha}^{x,y} \in F_{\alpha}$.
 - g. Ganti masing-masing bit pertama untuk tiap piksel dari FI dengan nilai F_{α} .
 - h. Lakukan proses a – g, hingga semua bit FI telah diganti, dengan catatan *natural image* yang digunakan adalah citra awal.
 - i. Lakukan proses h untuk semua *natural image*.
 - j. Jika semua bit telah diganti, maka *natural share image* telah dihasilkan.
 - k. Lakukan penggabungan dengan melakukan operasi XOR masing-masing nilai piksel *natural share image*.

3. Encryption

Lakukan operasi XOR nilai masing-masing nilai R, G dan B untuk masing-masing nilai piksel hasil gabungan *natural share image* dengan *secret image*.

Berikut ini merupakan *pseudocode* proses pembentukan *share* yang terdiri dari tiga subrutin yaitu (Liu et al., 2013):

1. Procedure FE (), *input* pada procedure ini adalah $(n - 1)$ *natural image* dan *output*-nya merupakan $(n - 1)$ *natural image* yang telah diacak.
2. Procedure FE_bin (), *input* pada procedure ini adalah $(n - 1)$ *natural image* yang telah diacak dan *output*-nya merupakan F_{α} yaitu citra biner dengan ukuran yang sama dengan *natural image*.
3. Encryption (), *input* pada procedure ini adalah *secret image* dan hasil gabungan $(n - 1)$ *natural share image* dan *output*-nya merupakan *share image*.

Procedure Pembentukan Share ():

Input: S, N_1, \dots, N_{n-1}

Output: $\bar{S}, FI_1, \dots, FI_{n-1}$

1. For $\alpha = 1$ to $(n - 1)$
2. For $\varphi \in \{R, G, B\}$
3. For $i = 1$ to 8
4. $FI_{\alpha,\varphi,i} = 0$
5. For $i = 1$ to 8
6. Procedure FE()
7. For $\alpha = 1$ to $(n - 1)$
8. $FI_{\alpha,R,i} = F_\alpha$
9. Procedure FE()
10. For $\alpha = 1$ to $(n - 1)$
11. $FI_{\alpha,G,i} = F_\alpha$
12. Procedure FE()
13. For $\alpha = 1$ to $(n - 1)$
14. $FI_{\alpha,B,i} = F_\alpha$
15. For $\varphi \in \{R, G, B\}$, $hasilMerge_\varphi = FI_{1,\varphi} \text{ XOR } FI_{2,\varphi} \text{ XOR } \dots \text{ XOR } FI_{n-1,\varphi}$
16. Procedure Encryption ():

Procedure FE ():

1. For $\alpha = 1$ to $(n - 1)$
2. For $\varphi \in \{R, G, B\}$
3. For $q = 1$ to t_p
 - 4. Ambil koordinat (x_1, y_1) secara acak, $x_1 \in [1, W]$, $y_1 \in [1, H]$
 - 5. Ambil koordinat (x_2, y_2) secara acak, $x_2 \in [1, W]$, $y_2 \in [1, H]$
 - 6. Tukar nilai $p_{\alpha,\varphi}^{x_1,y_1}$ dan $p_{\alpha,\varphi}^{x_2,y_2}$
7. For $\alpha = 1$ to $(n - 1)$
8. For $\varphi \in \{R, G, B\}$

```

9.           $p_{prev} = 0$ 
10.         For x = 1 to W
11.             For y = 1 to H
12.                  $p_{\alpha,\varphi}^{x,y} = p_{prev} \text{ XOR } p_{\alpha,\varphi}^{x,y}$ 
13.                  $p_{prev} = p_{\alpha,\varphi}^{x,y}$ 
14. Procedure FE_bin()

```

Procedure FE_bin ():

```

1.  For  $\alpha = 1$  to  $(n - 1)$ 
2.    Bagi  $N_\alpha$  menjadi  $t$  blok dengan  $8 \times 8$  piksel
3.    For  $\alpha = 1$  to  $(n - 1)$ 
4.      For  $\beta = 1$  to  $t$ 
5.        For  $x = x_\beta$  to  $(x_\beta + 7)$ 
6.          For  $y = y_\beta$  to  $(y_\beta + 7)$ 
7.             $H_\alpha^{x,y} = p_{\alpha,R}^{x,y} + p_{\alpha,G}^{x,y} + p_{\alpha,B}^{x,y}$ 
8.             $M_\alpha^\beta = \frac{H_\alpha^{x_\beta,y_\beta}, \dots, H_\alpha^{x_\beta+7,y_\beta+7}}{\text{total } H_\alpha^{x,y}}$ 

```

```

9.      For  $x = x_\beta$  to  $(x_\beta + 7)$ 
10.        For  $y = y_\beta$  to  $(y_\beta + 7)$ 
11.           $f_\alpha^{x,y} = \begin{cases} 1, & H_\alpha^{x,y} < M_\alpha^\beta \\ 0, & \text{jika tidak} \end{cases}$ 

```

Procedure Encryption ():

For $\varphi \in \{R, G, B\}$, $\bar{S}_\varphi = S_\varphi \text{ XOR } hasilMerge_\varphi$

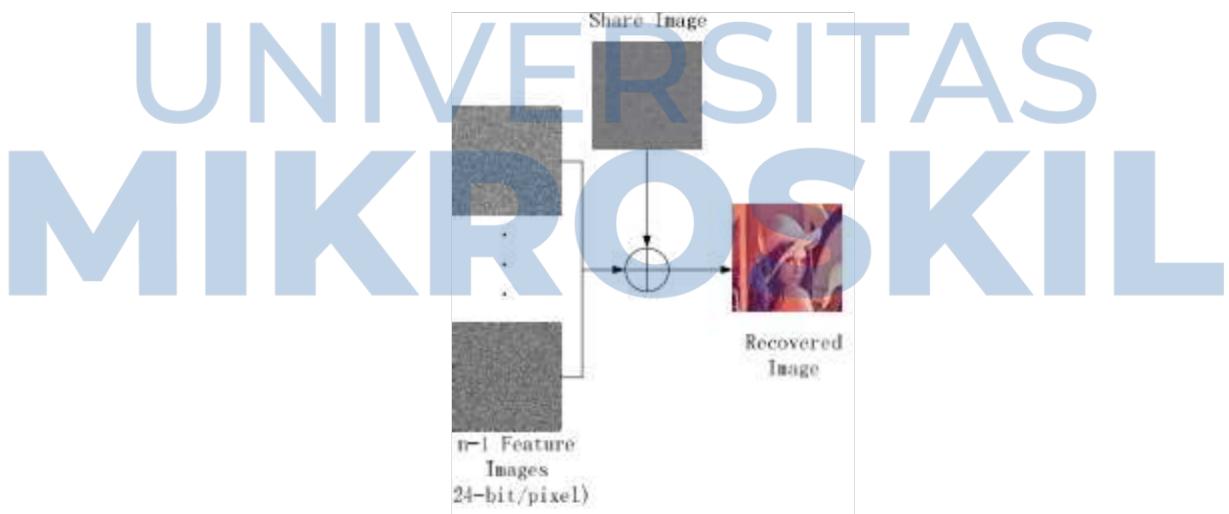
Dimana:

- S : citra *input*
- N_i : *natural image* ($1 \leq i \leq (n - 1)$)
- \bar{S} : citra *output*
- $H_\alpha^{x,y}$: asumsi nilai piksel RGB (x, y) pada *natural image* α (N_α)

- $p_{\alpha,\varphi}^{x,y}$: nilai piksel pada *natural image* α (N_α), dengan warna φ pada koordinat (x, y)
- M_α^β : rata-rata nilai piksel pada blok β ($H_\alpha^{x\beta,y\beta}, \dots, H_\alpha^{x\beta+7,y\beta+7}$)
- $f_\alpha^{x,y}$: nilai piksel untuk koordinat (x, y) pada *natural share image*, $f_\alpha^{x,y} \in F_\alpha$.
- S_φ : nilai φ pada S , dimana $\varphi \in \{R, G, B\}$
- \bar{S}_φ : nilai φ pada \bar{S} , dimana $\varphi \in \{R, G, B\}$
- FI_α : peta fitur gambar biner dari *natural image*
- $FI_{\alpha, \varphi}$: fitur gambar biner pada warna φ , dimana $\varphi \in \{R, G, B\}$
- $FI_{\alpha, \varphi, i}$: fitur gambar biner pada warna φ dan pada nilai i , $FI_{\alpha, \varphi, i} \in \{0, 1\}$

2.3.2. Proses Rekonstruksi *Share*

Proses rekonstruksi *share image* dilakukan dengan melakukan penggabungan $(n - 1)$ *natural share image*. Hasil gabungan kemudian dilakukan operasi XOR dengan *share image* sehingga menghasilkan *secret image* (Liu et al., 2013). Proses rekonstruksi dijelaskan pada Gambar 2. 20 berikut:



Gambar 2. 20 Proses Rekonstruksi

(Sumber: Liu et al., 2013)

Berikut ini merupakan *pseudocode* proses rekonstruksi *share* (Liu et al., 2013).

Procedure Rekonstruksi Share ():

Input: $\bar{S}, FI_1, \dots, FI_{n-1}$

Output: S

1. For $\varphi \in \{R, G, B\}$, $hasilMerge_\varphi = FI_{1,\varphi} \text{ XOR } FI_{2,\varphi} \text{ XOR } \dots \text{ XOR } FI_{n-1,\varphi}$
2. For $\varphi \in \{R, G, B\}$, $S_\varphi = \bar{S}_\varphi \text{ XOR } hasilMerge_\varphi$

2.4. Steganografi

Steganografi (steganography) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia. Kata steganografi berasal dari bahasa Yunani yang berarti “tulisan tersembunyi” (*covered writing*). Steganografi membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks atau video. Steganografi dapat dipandang sebagai kelanjutan kriptografi. Jika pada kriptografi, data yang telah disandikan (*ciphertext*) tetap tersedia, maka dengan steganografi ciphertext dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaanya. Di negara-negara yang melakukan penyensoran informasi, steganografi sering digunakan untuk menyembunyikan pesan-pesan melalui gambar (*image*), video atau suara (*audio*) (Munir, 2004).

Kriteria yang harus diperhatikan dalam steganografi adalah (Munir, 2004):

1. Fidelity

Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.

2. Robustness

Data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung (seperti pengubahan kontras, penajaman, pemampatan,

rotasi, perbesaran gambar, pemotongan (*cropping*), enkripsi dan sebagainya. Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak.

3. Recovery

Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*). Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

Dalam steganografi, metode yang umum digunakan adalah *Least Significant Bit (LSB)*. LSB adalah metode steganografi dengan menyisipkan data rahasia dalam bentuk bit pada bit rendah atau bit yang paling kanan (LSB) pada data piksel yang menyusun file pembawa. Misalnya pada media pembawa berupa citra bitmap, pada berkas bitmap 24 bit, setiap piksel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian, pada setiap piksel berkas bitmap 24 bit kita dapat menyisipkan 3 bit data (Krisnawati, 2008).

Kekurangan dari LSB adalah dapat secara drastis mengubah unsur pokok warna dari piksel. Ini dapat menunjukkan perbedaan yang nyata dari *cover image* menjadi *stego image*, sehingga tanda tersebut menunjukkan keadaan dari steganografi. Variasi warna kurang jelas dengan 24 bit *image*, bagaimanapun file tersebut sangatlah besar. Antara 8 bit dan 24 bit *image* mudah diserang dalam pemrosesan *image*, seperti pemotongan (*cropping*) dan kompresi. Keuntungan dari LSB adalah cepat dan mudah dalam prosesnya (Krisnawati, 2008).

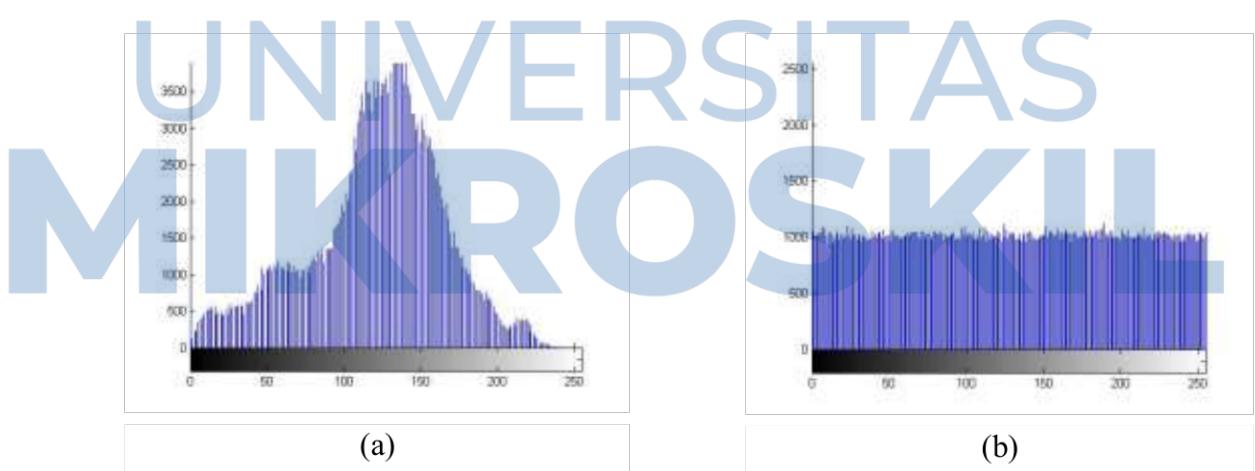
2.5. Analisis Data

Untuk mengolah data menjadi informasi, sehingga karakteristik sifat-sifat datanya dapat dengan mudah dipahami dan bermanfaat untuk kepentingan lainnya maka dilakukan penganalisaan data seperti analisis korelasi, analisis histogram, dan melihat nilai MSE dan PSNR (Munir, 2012).

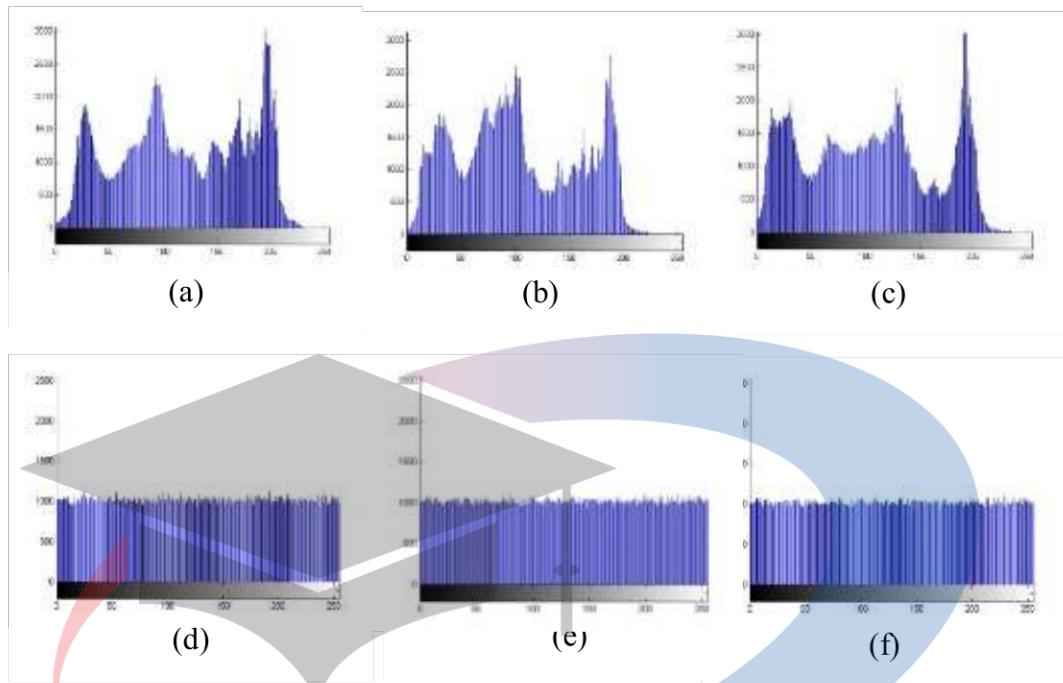
2.5.1. Analisis Histogram

Histogram merupakan salah satu fitur yang penting, sebab sebuah histogram memperlihatkan distribusi intensitas piksel-piksel di dalam citra tersebut. Dalam melakukan serangan dengan teknika analisis statistik, penyerang menggunakan histogram untuk menganalisis frekuensi kemunculan intensitas piksel untuk mendeduksi kunci atau piksel-piksel di dalam *plain-image*. Agar serangan dengan analisis statistik tidak dimungkinkan, maka di dalam enkripsi citra penting untuk menghasilkan histogram *cipher-image* yang tidak memiliki kemiripan secara statistik dengan histogram *plain-image*. Oleh karena itu, piksel-piksel di dalam *cipher-image* seharusnya memiliki distribusi yang (relatif) *uniform* atau ditunjukkan dengan histogram yang terlihat datar (*flat*) (Munir, 2012).

Gambar 2. 21 (a) memperlihatkan histogram citra ‘couple’ dan Gambar 2. 21 (b) adalah histogram *cipher-image*-nya. Histogram *cipher-image* terlihat datar dan berbeda dengan histogram *plain-image* (Munir, 2012).



Gambar 2. 21 (a) historam citra ‘couple’ (*plain-image*), (b) histogram *cipher-image*
(Sumber: Munir, 2012)



Gambar 2.22 (a)-(c) Histogram citra ‘yacht’ (*plain-image*) pada kanal RGB; dan (d)-(f) histogram *cipher-image* untuk setiap kanal
(Sumber: Munir, 2012)

Seperti pada citra ‘couple’, histogram setiap kanal RGB pada *cipher-image* juga berbentuk *flat* atau terdistribusi *uniform* (Munir, 2012).

2.5.2. Analisis Korelasi

Analisis korelasi merupakan studi yang membahas tentang derajat (seberapa kuat) hubungan antara dua variabel atau lebih. Ukuran derajat hubungan disebut koefisien korelasi. Koefisien korelasi merupakan angka yang menunjukkan arah dan kuat hubungan antara dua variabel atau lebih. Koefisien korelasi (r_{xy}) dari dua buah peubah acak diskrit yang masing-masing beranggotakan n elemen dihitung dengan rumus berikut (Munir, 2012):

yang dalam hal ini

Arah dinyatakan dalam bentuk hubungan positif atau negatif. Arah hubungan positif berarti apabila nilai variabel ditingkatkan maka akan meningkatkan nilai variabel yang lain, dan sebaliknya apabila nilai variabel diturunkan maka akan menurunkan nilai variabel yang lain. Arah hubungan negatif berarti apabila nilai variabel ditingkatkan maka akan menurunkan nilai variabel lain, dan sebaliknya apabila nilai variabel diturunkan maka akan meningkatkan nilai variabel lain. Kuat hubungan dinyatakan dalam bentuk angka, antara 0-1. Angka 0 menunjukkan hubungan yang tidak ada, sementara angka 1 menunjukkan hubungan yang sempurna. Semakin kecil koefisien korelasi, maka akan semakin besar kesalahan untuk membuat prediksi. Besar koefisien korelasi memiliki range dari -1 sampai +1. Nilai koefisien korelasi +1 menyatakan hubungan korelasi sempurna yang menaik, nilai koefisien korelasi -1 menyatakan hubungan koreasi sempurna yang menurun, sedangkan antara -1 dan +1 menyatakan derajat ketergantungan korelasi antara dua peubah. Nilai koefisien yang dekat dengan -1 atau +1 menyatakan hubungan korelasi yang kuat antara x dan y, sedangkan nilai koefisien yang dekat dengan 0 menyatakan hubungan korelasi yang lemah. Besar koefisien korelasi dapat diketahui berdasarkan penyebaran titik pertemuan antara dua variabel (Marhaendro, 2012). Adapun nilai koefisien korelasi dan interpretasinya ditunjukkan pada Tabel 2.1 berikut (Munir, 2012):

Tabel 2. 1 Koefisien Korelasi Dan Interpretasinya

Nilai korelasi (r)	Interpretasinya
0,00 – 0,199	Sangat rendah
0,20 – 0,399	Rendah
0,40 – 0,599	Sedang
0,60 – 0,799	Kuat
0,80 – 1,000	Sangat kuat

Pada kebanyakan citra *plain-image*, koefisien korelasi antara piksel-piksel bertetangga biasanya tinggi (mendekati +1 atau -1). Tujuan enkripsi citra adalah menghilangkan korelasi antara piksel-piksel tersebut atau membuat koefisien korelasinya mendekati nol. Untuk menyelidiki korelasi pada *plain-image* dan *cipher-image*, maka dihitung korelasi antara dua piksel yang bertetangga secara horizontal [$f(i,j)$ dan $f(i,j+1)$], dua piksel yang bertetangga secara vertikal [$f(i,j)$ dan $f(i+1,j)$], dan dua piksel yang bertetangga secara diagonal [$f(i,j)$ dan $f(i+1,j+1)$], baik pada *plain-image* maupun pada *cipher-image* (Munir, 2012).

Munir et al. melakukan eksperimen menggunakan 1000 pasang piksel bertetangga yang dipilih secara acak pada setiap arah (vertikal, horizontal, dan diagonal) dari citra ‘couple’ beserta citra hasil enkripsinya. Koefisien korelasi dihitung dengan persamaan (13), yang dalam hal ini x dan y adalah nilai keabuan dari dua piksel bertetangga. Hasil eksperimen diperlihatkan pada Tabel 2. 2 berikut ini (Munir, 2012):

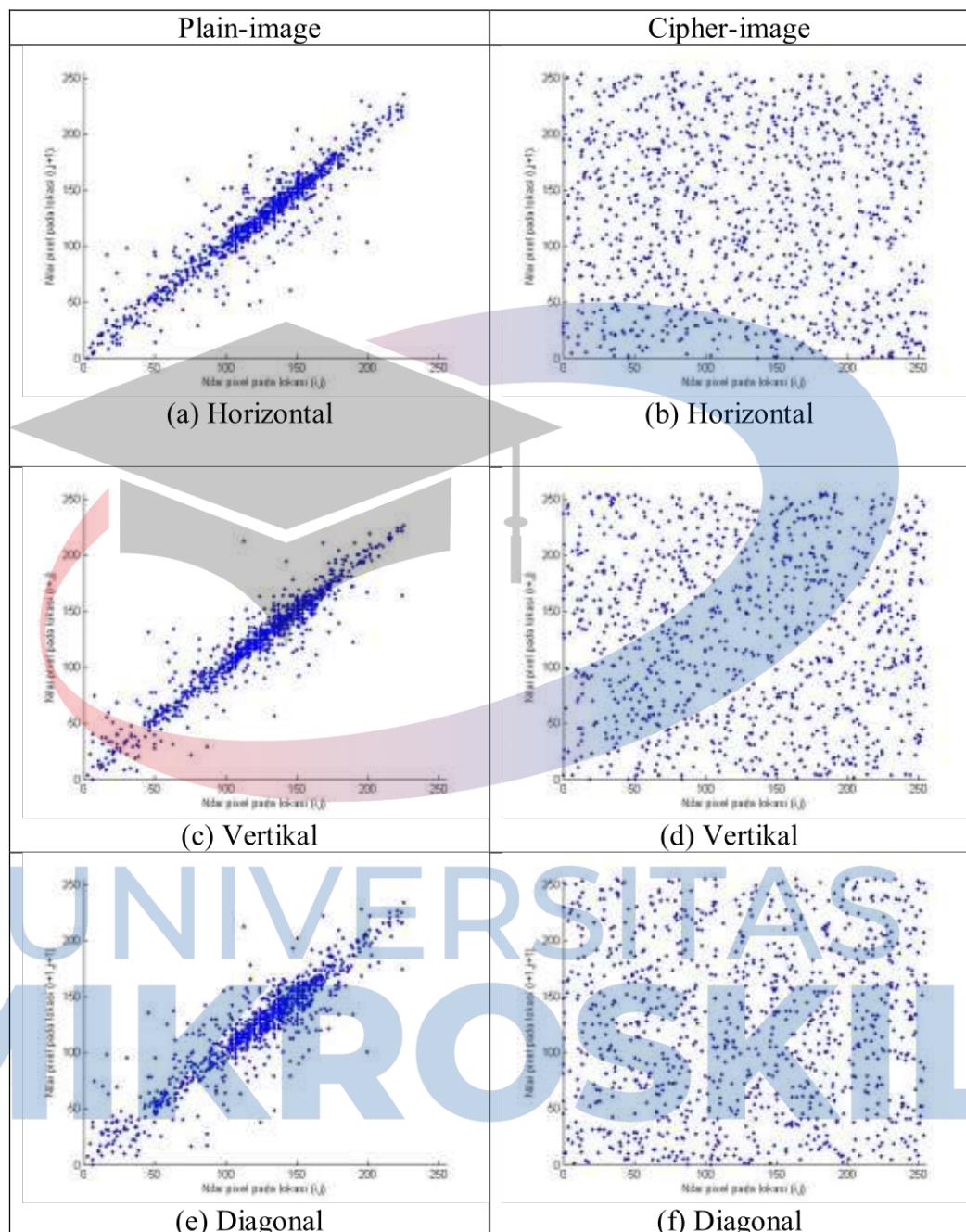
Tabel 2. 2 Perbandingan koefisien korelasi antara dua piksel bertetangga

Koefisien Korelasi	Horizontal	Vertikal	Diagonal
<i>Plain-image</i>	0.9364	0.9422	0.9046
<i>Cipher-image</i>	-0.0142	0.0244	0.0471

Dari Tabel 2. 2 dapat dilihat bahwa koefisien korelasi pada piksel-piksel bertetangga pada setiap arah di dalam *plain-image* nilainya berada di sekitar angka 1, yang mengindikasikan korelasi yang kuat diantara piksel-piksel tersebut, tetapi pada *cipher-image* koefisien korelasi mendekati nol, yang mengindikasikan piksel-piksel yang bertetangga tidak lagi berkorelasi (Munir, 2012).

Untuk memperlihatkan situasi yang lebih jelas mengenai korelasi, Tabel 2. 3 memperlihatkan distribusi korelasi piksel-piksel yang bertetangga pada masing-masing *plain-image* (kolom kiri) dan *cipher-image* (kolom kanan) (Munir, 2012).

Tabel 2. 3 Distribusi korelasi piksel bertetangga *plain-image* dan *cipher-image*



Pada *plain-image*, piksel-piksel yang bertetangga nilainya berada di sekitar garis diagonal 45° , yang mengindikasikan korelasi yang kuat antara piksel-piksel tersebut. Sebaliknya, pada *cipher-image* nilai-nilai piksel tersebar merata di seluruh area bidang datar, yang mengindikasikan piksel-piksel di dalamnya tidak lagi berkorelasi (Munir, 2012).

2.5.3. Mean Square Error dan Peak Signal to Noise Ratio

Dalam citra digital terdapat suatu standar pengukuran *error* (galat) kualitas citra, yaitu besaran PSNR dan MSE. Tingkat keberhasilan dan performa dari suatu metode *filtering* pada citra dihitung dengan menggunakan *Peak Signal to Noise Ratio* atau biasa disingkat PSNR. Meskipun performa metode *filtering* juga dapat diukur dengan teknik visual (hanya melihat pada citra hasil dan membandingkannya dengan citra yang terdapat *noise*). Namun hasil pengukuran teknik visual setiap manusia berbeda-beda. Sehingga MSE dan PNSR merupakan solusi pengukuran performa yang baik (Listiyani, 2007).

Peak Signal to Noise Ratio (PSNR) adalah sebuah perhitungan yang menentukan nilai dari sebuah citra yang dihasilkan. Satuan nilai dari PSNR sama seperti MSE, yaitu decibel (dB). Nilai PSNR ditentukan oleh besar atau kecilnya nilai MSE yang terjadi pada citra. Semakin besar nilai PSNR, semakin baik pula hasil yang diperoleh pada tampilan citra hasil dan hampir sama dengan citra aslinya. Sebaliknya, semakin kecil nilai PSNR, maka akan semakin buruk pula hasil yang diperoleh pada tampilan citra hasil. 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. Jadi hubungan antara nilai PSNR dengan nilai MSE adalah semakin besar nilai PSNR, maka akan semakin kecil nilai MSE-nya. PSNR secara umum digunakan untuk mengukur kualitas pada penyusunan ulang citra. Hal ini lebih mudah didefinisikan dengan *Mean Square Error* (MSE) (Listiyani, 2007).

Mean Square Error (MSE) adalah kesalahan kuadrat rata-rata. Nilai MSE didapat dengan membandingkan nilai selisih piksel-piksel citra asal dengan citra hasil pada posisi yang sama. Semakin besar nilai MSE, maka tampilan pada citra hasil akan semakin buruk. Sebaliknya, semakin kecil nilai MSE, maka tampilan pada citra hasil akan semakin baik (Listiyani, 2007).

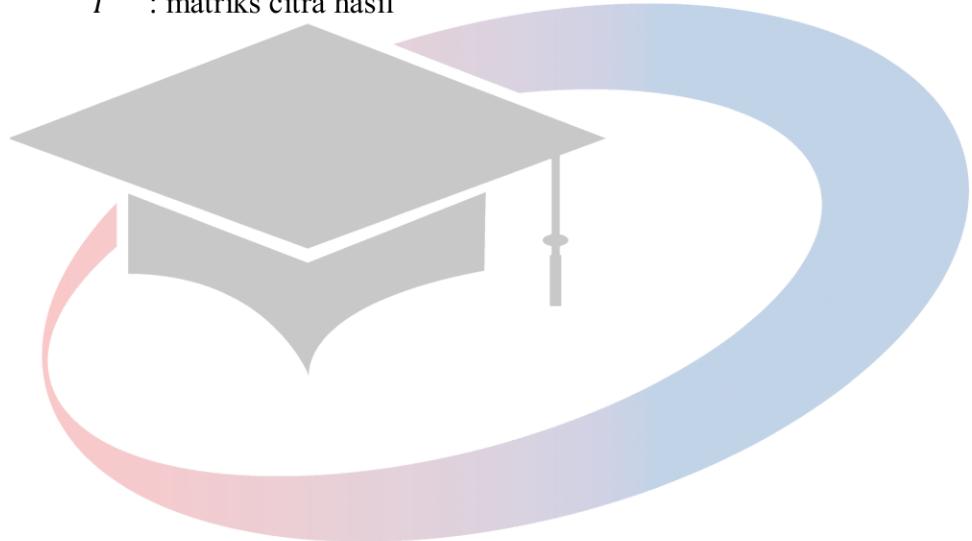
Misal $I(x, y)$ adalah citra masukan $I'(x, y)$ adalah citra keluaran, keduanya memiliki M baris dan N kolom, maka didefinisikan sebagai berikut (Listiyani, 2007):

Rumus untuk menghitung PSNR adalah (Listiyani, 2007):

$$PSNR = 20 \times \log_{10} \left(255 / \sqrt{MSE} \right) \dots \dots \dots \quad (19)$$

Dimana:

- x : ukuran baris dari citra
 - y : ukuran kolom dari citra
 - I : matriks citra awal
 - I' : matriks citra hasil



UNIVERSITAS MIKROSKIL