



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

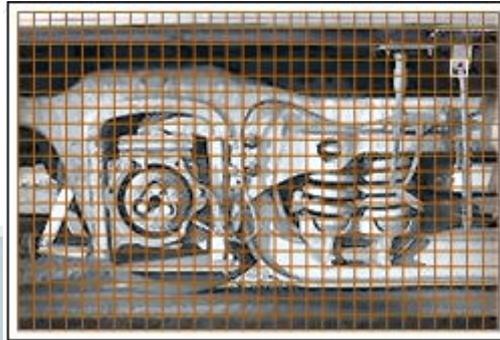
## BAB II

### LANDASAN TEORI

#### 2.1 Citra Digital

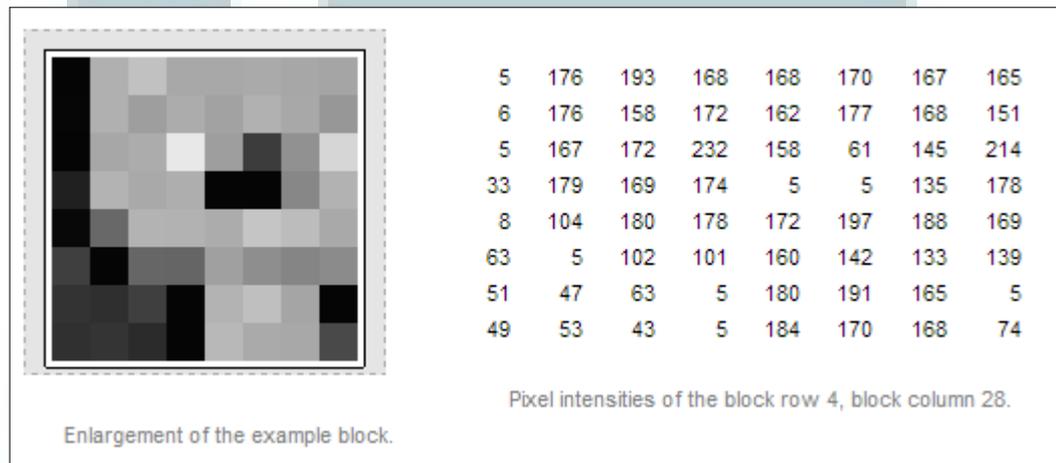
Kamus Besar Bahasa Indonesia menerjemahkan citra atau *image* sebagai gambar pada bidang dua dimensi. Ditinjau dari sudut pandang sains, citra tercipta karena adanya sumber cahaya yang menerangi sebuah objek, kemudian objek memantulkan kembali sebagian cahaya tersebut. Cahaya yang dipantulkan inilah yang kemudian ditangkap oleh optik atau panca indera manusia sehingga bayangan objek tersebut terekam. Apabila bayangan objek tersebut ditangkap oleh mesin digital maka citra tersebut dapat digolongkan dalam kategori citra digital (Sada, 2011).

Pada komputer, citra digital direpresentasikan menggunakan susunan persegi suatu piksel (Sachs, 1996). Piksel menggambarkan tingkat keabuan pada suatu gambar hitam putih dan merupakan satuan numerik yang merepresentasikan titik terkecil pada gambar. Citra digital terdapat dua jenis yakni citra dengan kompresi *lossy* dan *lossless*. Kompresi *lossy* adalah kompresi pada gambar yang menggunakan teknik DCT (*Discrete Cosine Transform*) dan tabel kuantisasi (Sistem Multimedia, 2011). Pemrosesan DCT dilakukan pada setiap blok piksel pada citra yang berukuran 8 x 8. Proses partisi dapat dilihat pada gambar 2.1.



Gambar 2.1 Partisi gambar menjadi blok 8 x 8  
(Sumber: [www.whymath.org](http://www.whymath.org))

Selanjutnya citra akan diproses blok per blok dengan ukuran 8 x 8. Proses ini berjalan secara independen, maksudnya, blok satu tidak mempengaruhi nilai blok lainnya.



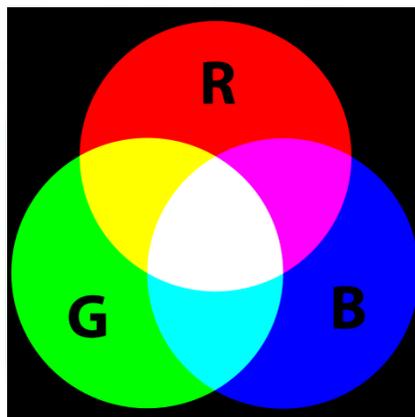
Gambar 2.2 Tampilan blok gambar beserta nilai pikselnya  
(Sumber: [www.whymath.org](http://www.whymath.org))

Proses DCT dan kuantisasi yang dialami sebuah citra menghasilkan efek *blockiness* (kotak-kotak) dan *blur* (pudar) pada gambar yang terkompres. *Blockiness* disebabkan oleh nilai diskontinu antar blok piksel yang dikarenakan proses kuantisasi yang dilakukan secara independen, sedangkan efek blur

disebabkan oleh banyaknya nilai koefisien DCT yang hilang sehingga memperhalus warna pada blok piksel (Wang, 2008). Contoh format citra yang mengalami kompresi *lossy* adalah JPG dan JPEG. Pada kompresi *lossless*, citra dikompresi dengan menggunakan metode prediksi yang dirujuk pada citra aslinya, dengan begitu tidak banyak informasi yang terhilang. Contoh format citra yang mengalami kompresi *lossless* yakni TIFF (*Tagged Image File Format*).

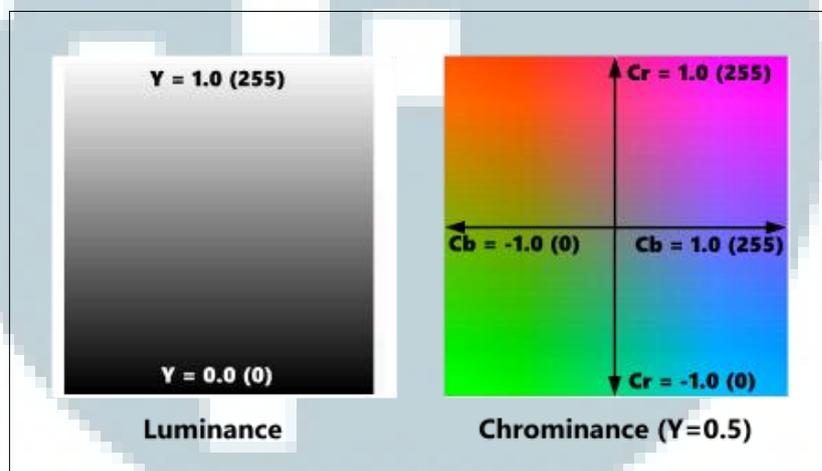
### 2.3 Komponen Warna

Komponen warna adalah model matematika yang mempresentasikan bilangan numerik dengan sebuah warna. Komponen warna yang digunakan pada citra digital pada umumnya adalah RGB (*Red Green Blue*). RGB merupakan sistem warna aditif dimana untuk menciptakan sebuah warna baru maka diperlukan adanya kombinasi antar warna antara sinar merah, hijau, dan biru.



Gambar 2. 3 Model warna RGB  
(Sumber: [www.printingwiki.org](http://www.printingwiki.org))

Komponen warna YCbCr merupakan model warna yang mempresentasikan citra menggunakan *luminance* (Y) dan *chrominance* (Cb dan Cr). Luminance adalah intensitas cahaya pada gambar, Cb adalah *blue difference* atau perbedaan komponen warna biru dan Cr merupakan *red difference* atau perbedaan warna merah pada komponen warna. Model YCbCr digunakan untuk merepresentasikan informasi RGB pada suatu gambar.



Gambar 2.4 Komponen warna YCbCr  
(Sumber: msdn.microsoft.com)

Citra yang ingin dilakukan proses kompresi sebelumnya harus diubah dahulu komponen warnanya dari RGB ke YCbCr. Pada model warna YCbCr nilai *luminance* dan *chrominance* disimpan pada nilai yang berbeda, oleh karena itu mempermudah proses manipulasi pada gambar. Proses pengkonversian model RGB menuju YCbCr dapat dihitung menggunakan rumus 2.1 sebagai berikut,

$$[Y \ Cb \ Cr] = [R \ G \ B] \begin{bmatrix} 0.299 & -0.168935 & 0.499813 \\ 0.587 & -0.331665 & -0.418531 \\ 0.114 & 0.50059 & -0.081282 \end{bmatrix}$$

.....Rumus 2.1

Dimana Y merupakan *luminance*, Cb dan Cr merupakan *chrominance*, dan R,G,B berturut – turut adalah nilai dari sinar merah (*Red*), sinar hijau (*Green*), dan sinar biru (*Blue*).

### 2.3 Image Forgery

*Image forgery* atau pemalsuan citra adalah suatu kegiatan memodifikasi citra sehingga berbeda dari tampak awal citra yang aslinya. Pemalsuan citra dapat dilakukan dengan beberapa metode. Menurut Mahdian, metode - metode yang termasuk dalam kegiatan pemalsuan citra yakni, menghapus atau menyembunyikan bagian gambar, menambah objek baru pada gambar, dan membuat salah tafsir informasi pada gambar. Metode - metode tersebut dapat dicapai dengan teknik modifikasi citra yakni teknik *copy-move* atau *image splicing*. Pada metode *copy-move* bagian tertentu pada suatu gambar digandakan sehingga ada beberapa bagian pada gambar tersebut yang sama atau identik. *Image splicing* adalah kegiatan mengambil sample dari satu gambar lalu meletakkannya ke dalam gambar lain. Untuk mengetahui apakah sebuah gambar telah mengalami pemalsuan atau tidak maka diperlukan adanya beberapa langkah penafsiran kualitas citra (Hsu, 2008). Langkah-langkah penafsiran citra yang menurut Hsu perlu dilakukan antara lain pengecekan *blocking*, *ringing*, dan

*blurring* pada gambar. Seperti yang telah dijelaskan sebelumnya, efek *blocking* terjadi karena adanya proses kuantisasi pada citra yang dilakukan secara independen pada suatu blok citra sehingga menghasilkan warna yang diskontinu antar *block boundaries*. Efek *blocking* ini memberikan kesan adanya *noise* pada suatu gambar atau biasa disebut dengan istilah pecah.



Gambar 2.5 Blockiness pada gambar  
(Sumber: [www.m.eet.com](http://www.m.eet.com))

*Blurring* adalah distorsi spasial yang menyebabkan kurangnya detil dan ketajaman pada gambar. Efek *blur* dapat tercipta baik sengaja baik tidak disengaja. Efek *blur* yang tidak sengaja tercipta diakibatkan oleh pengambilan foto yang tidak stabil sedangkan efek *blur* yang disengaja disebabkan oleh proses penghalusan pada citra menggunakan piranti lunak tertentu. Pada proses kompresi citra JPEG, efek *blurring* dapat terjadi karena banyaknya nilai koefisien DCT yang hilang selama proses kompresi citra terjadi sehingga memperhalus sinyal gambar pada setiap bloknya.



Gambar 2.6 Efek *blur* pada gambar  
(Sumber: <http://wparchive.risd.edu>)

Ringing adalah sebuah distorsi citra yang membuat sebuah efek lingkaran pada tepi gambar. Hal ini terjadi karena adanya perbedaan yang mencolok pada ketajaman dan warna pada bagian gambar.



Gambar 2.7 *Ring effect* pada bagian muka gambar  
(Sumber: Google Image)

Namun pengecekan kualitas citra sangat bergantung pada ketersediaan dokumen aslinya agar kegiatan perbandingan antara dua gambar dapat dilakukan (Wang, 2004). Adapun jenis pengecekan berdasarkan ketersediaan dokumen asli dibedakan menjadi tiga jenis yakni:

- *Full referenced method* yakni kegiatan membandingkan antar dua gambar yakni gambar asli dengan gambar yang telah dimodifikasi. Masalahnya adalah kerap kali gambar asli tidak tersedia sehingga metode ini tidak efektif.
- *Reduced referenced metrics* yakni kegiatan membandingkan sebagian dari gambar asli dengan bagian pada gambar yang dianggap palsu. Sama seperti *full referenced method*, metode ini bergantung pada ketersediaan gambar aslinya.
- *No-referenced metrics* yakni kegiatan pengecekan gambar apakah asli atau sudah termodifikasi tanpa diperlukan adanya dokumen asli. Metode ini sangat efektif mengingat gambar atau dokumen asli sulit didapatkan. Pengecekan menggunakan metode ini bertumpu pada langkah-langkah penafsiran yang dipaparkan oleh Hsu.

Selain metode pengecekan pemalsuan gambar yang telah dilansir oleh Hsu, menurut Mahdian perlu adanya analisis yang lebih dalam pemeriksaan citra yang dipalsukan. Adapun metode yang diajukan adalah sebagai tolak ukur tambahan untuk memastikan benar adanya kegiatan pemalsuan pada gambar. Salah satu metode yang diajukan oleh Mahdian adalah pemeriksaan *noise inconsistencies* pada gambar. Biasanya jumlah noise pada suatu gambar unik antara satu dengan yang lainnya. Dengan menambah gambar lainnya atau melakukan kegiatan *image splicing* tentunya akan membuat *noise* pada gambar menjadi tidak seragam. Oleh karena itu, perbedaan noise pada gambar dapat mengindikasikan bahwa adanya modifikasi pada gambar di daerah terdapatnya *noise* yang tidak konsisten.

## 2.4 No-reference Quality Metrics

*No-reference quality metrics* adalah sebuah metode yang diciptakan oleh Wang et al. untuk memeriksa adanya penurunan kualitas pada gambar atau citra tanpa memerlukan adanya ketersediaan citra asli sebagai pembandingnya.. Sebagai gantinya metode ini memerlukan beberapa tahapan atau langkah untuk melakukan pemeriksaan terhadap inkonsistensi pada gambar. Langkah – langkah yang dilakukan antara lain *blocking*, *activity* dan *zero crossing*.

### 2.4.1 Blocking

Citra dengan format JPEG melalui proses kompresi berbasis blok dengan ukuran per bloknya 8 x 8 piksel. Pada saat proses kompresi berlangsung gambar mengalami proses *blurring* dan *blocking*. Proses *blurring* terjadi akibat hilangnya atau berkurangnya frekuensi koefisien pada *Discrete Cosine Transform* (DCT) sehingga memperhalus signal citra disetiap bloknya (Wang, 2008). Sedangkan *blocking* terjadi karena adanya blok-blok yang bersifat diskontinu yang disebabkan oleh proses kompresi pada JPEG karena pemrosesan dilakukan secara independen disetiap blok-blok. Pada langkah ini citra akan dihitung rata-rata perbedaan antar bloknya ( $B_h$ ) disepanjang garis horizontal pada gambar  $x$  yang berukuran  $M \times N$ . Berikut adalah formula perhitungan *blocking* pada gambar:

$$B_h = \frac{1}{M(\lfloor N/8 \rfloor - 1)} \sum_{i=1}^M \sum_{j=1}^{\lfloor N/8 \rfloor - 1} |d_h(i, 8j)|$$

.....Rumus 2.2

dimana  $M$  dan  $N$  berturut-turut adalah lebar dan tinggi gambar.  $D_h(m,n)$  adalah nilai selisih dari dua piksel yang saling bersebelahan. Nilai  $D_h(m,n)$  yang dapat dicari dengan menggunakan formula

$$d_h(m, n) = x(m, n + 1) - x(m, n), \quad n \in [1, N - 1] \quad \text{.....Rumus 2.3}$$

dimana  $m$  dan  $n$  merupakan letak baris dan kolom piksel yang ingin dihitung nilai selisihnya. Total nilai  $D_h$  akan menghasilkan nilai koefisien *sum of absolute differences* (SAD) pada gambar. Menurut Ahuja (2010), SAD merupakan teknik yang dipakai untuk mengetahui korelasi piksel antar dua citra. Nilai yang dihasilkan oleh SAD akan menghasilkan citra ketiga yang menggambarkan perbedaan antar dua citra yang dibandingkan.

#### 2.4.2 Activity

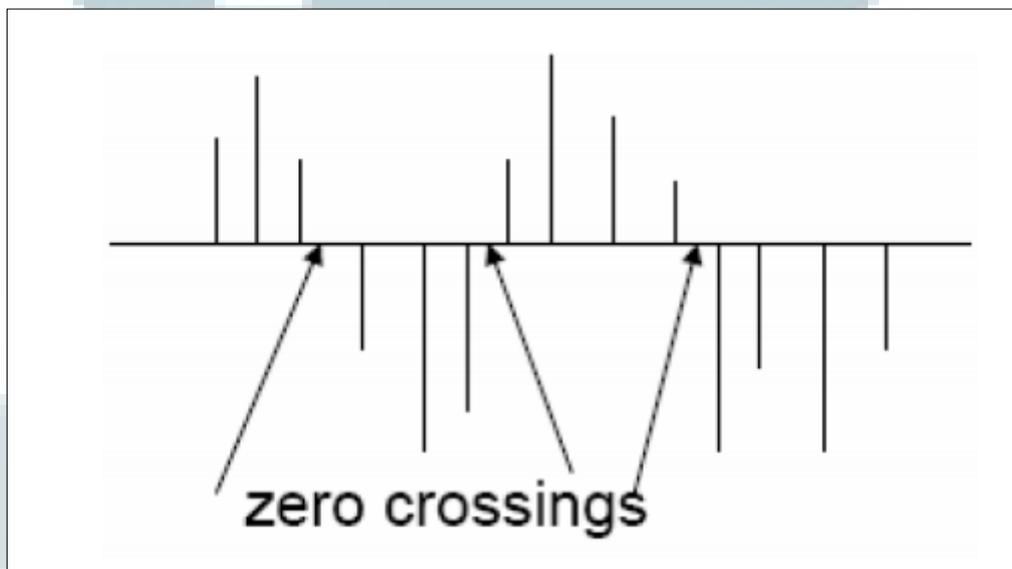
Efek *blurring* pada citra yang dihasilkan dari proses kompresi JPEG sulit untuk dideteksi apabila tidak ada citra aslinya sebagai pembanding. Oleh karena itu, menurut Wang (2008, pg. 3), untuk mengetahui adanya proses *blurring* yang terjadi pada citra diperlukan adanya perhitungan mengenai aktifitas piksel. Efek *blurring*, seperti yang sudah dijelaskan sebelumnya, terjadi karena kecilnya koefisien DCT yang terjadi oleh kompresi JPEG. Untuk itu, *blurring* dideteksi apabila adanya penurunan aktifitas piksel pada blok citra. Perhitungan aktifitas piksel dapat dihitung menggunakan formula sebagai berikut:

$$A_h = \frac{1}{7} \left[ \frac{8}{M(N-1)} \sum_{i=1}^M \sum_{j=1}^{N-1} |d_h(i, j)| - B_h \right] \quad \text{.....Rumus 2.4}$$

Dimana  $A_h$  adalah nilai aktifitas piksel disepanjang garis horizontal pada citra yang berukuran  $M \times N$ ,  $D_h$  merupakan nilai SAD dan  $B_h$  adalah nilai *blocking* disepanjang garis horizontal.

### 2.4.3 Zero Crossing

Pada *digital image processing*, *zero crossing* digunakan sebagai metode untuk memeriksa tepi pada suatu gambar dan dikenal juga sebagai metode *edge detection* atau *gradient filter*. Gradient filter adalah sebuah filter yang mencari area yang memiliki perbedaan nilai piksel yang mencolok. Ide dasar dari *zero crossing* adalah mencari nilai signal digital pada citra yang telah nilainya telah melewati (*cross*) nilai 0, baik nilai yang bersifat positif maupun negatif.



Gambar 2.8 *Zero Crossing*  
(Sumber: NAS, pp. 6)

Titik nilai signal yang melewati nilai batas 0 tersebut merupakan titik yang diindikasikan dimana perbedaan nilai signal sangat mencolok, sehingga dapat

disimpulkan bahwa titik tersebut merupakan titik potensi adanya *edge* atau tepi. Pada penelitian ini, *zero crossing* dihitung menggunakan nilai yang didapat dari perhitungan SAD pada tiap piksel. Perhitungan *zero crossing rate* pada citra dihitung secara horizontal menggunakan formula berikut ini:

$$Z_h = \frac{1}{M(N-2)} \sum_{i=1}^M \sum_{j=1}^{N-2} z_h(m, n),$$

.....Rumus 2.5

dimana

$$z_h(m, n) = \begin{cases} 1 & \text{horizontal Z at } d_h(m, n) \\ 0 & \text{otherwise} \end{cases}$$

dengan nilai  $Z_h = 1$  jika nilai selisih piksel tidak sama dengan 0 dan akan bernilai 0 jika nilai selisih piksel lebih dari 0. Apabila rate yang dihasilkan tinggi maka diindikasikan terdapat banyak *edge* disepanjang garis horizontal dan sebaliknya.

## 2.5 Sum of Absolute Difference atau SAD

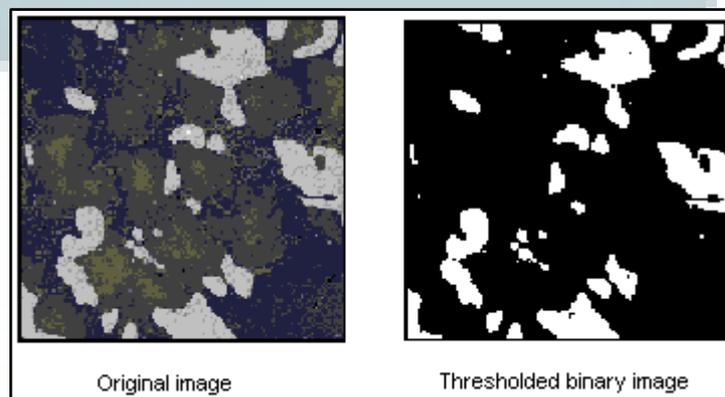
Sum of absolute difference adalah algoritma yang digunakan untuk mencari kemiripan (*similarity*) blok piksel pada gambar. Untuk mendapatkan nilai SAD, maka setiap piksel pada gambar akan dikurangi dengan gambar yang dibandingkan kemudian nilai diabsolutkan untuk menghindari adanya nilai yang bernilai negatif. Nilai – nilai yang didapatkan dijumlah (*sum*) untuk mendapatkan nilai kemiripan antar dua gambar yang bersangkutan. Adapun formula perhitungan SAD adalah sebagai berikut:

$$\sum_{(i,j) \in W} |I_1(i,j) - I_2(x+i, y+j)| \dots\dots\dots \text{Rumus 2.6}$$

dimana I adalah *image* atau citra i dan j merupakan letak piksel pada gambar, sedangkan x dan y merupakan letak piksel lainnya.

## 2.6 Thresholding

Thresholding pada *image processing* adalah metode yang digunakan untuk mensegmentasi atau mengelompokkan gambar menjadi bagian – bagian berdasarkan intensitas warnanya. Pada citra *grayscale* atau berwarna hitam putih, metode ini dapat menciptakan gambar biner (Shapiro, 2001). Gambar yang mendapat efek threshold dapat dilihat pada gambar 2.8.



Gambar 2.9 Contoh citra dengan efek *threshold*  
(Sumber: [www.ni.com](http://www.ni.com))

## 2.7 Korelasi

Dalam ilmu statistik, korelasi adalah nilai yang menggambarkan keterkaitan atau ketergantungan antara dua buah nilai set data. Rumus perhitungan yang umum dipakai dalam menghitung korelasi yakni *Pearson Product Moment Coefficient* atau disingkat *correlation coefficient*. Adapun rumus koefisien korelasi adalah sebagai berikut:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \dots\dots\dots \text{Rumus 2.7}$$

dimana  $R_{xy}$  merupakan nilai koefisien korelasi antara set data x dan set data y, serta  $\bar{X}$  bar dan  $\bar{Y}$  bar masing-masing merupakan nilai rata-rata dari masing-masing set data. Jika koefisien yang didapat bernilai 0 maka antar kedua buah set data tidak ditemukan adanya kecocokan. Sebaliknya, jika koefisien bernilai 1 maka ditemukan adanya kecocokan antar kedua buah set data.

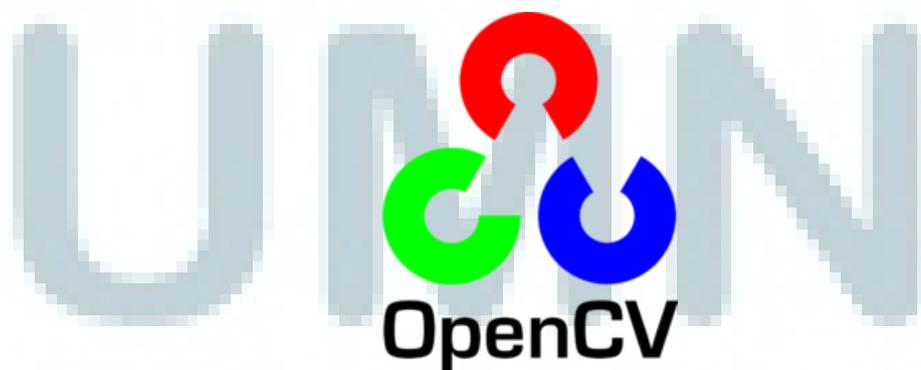
## 2.8 Microsoft Visual C#

Microsoft Visual C# adalah *Integrated Development Environment* (IDE) yang dikembangkan oleh Microsoft untuk membantu *programmer* dalam mengembangkan aplikasi dengan menggunakan bahasa pemrograman C# dengan menggunakan *framework* yang sudah disediakan oleh Microsoft sendiri yaitu .Net (baca: dot net). Microsoft Visual C# memudahkan programmer untuk menciptakan piranti lunak baik *desktop*, *web based*, maupun *mobile*. Keunggulan

daripada IDE ini adalah kemudahan yang ditawarkan untuk membuat *user interface* hanya dengan cara *drag and drop*, serta *object refactoring* yang dapat menulis secara otomatis *constructor* dan *destructor*, serta fungsi - fungsi yang terdapat pada objek yang telah didefinisikan oleh programmer, sehingga Microsoft Visual C# menyandang predikat *rapid application development*. C# sendiri adalah bahasa pemrograman yang berorientasi objek. Pada saat ini versi Microsoft Visual C# sudah sampai dengan 5.0 yang dapat digunakan dengan IDE Microsoft Visual Studio 2012. Versi ini sudah mendukung pengembangan perangkat *mobile* dengan sistem operasi Windows Phone.

## 2.9 EmguCv

EmguCV adalah wrapper untuk OpenCV sehingga memungkinkan untuk dipakai pada bahasa pemrograman C#. OpenCV atau *Open Source Computer Vision Library* adalah library yang dikembangkan oleh Intel dengan tujuan untuk memudahkan programmer melakukan kegiatan manipulasi dan pengembangan aplikasi yang berhubungan dengan video dan gambar.



Gambar 2.10 Lambang OpenCV  
(Sumber: [www.opencv.org](http://www.opencv.org))