

# PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB

**RD. Kusumanto, Alan Novi Tomponu**

*Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang 30139  
Jl Sriwijaya Negara Bukit Besar, Palembang 30139*

*Handphone: 0812-7853505, 0812-3362210, Phone: +62-711-353414. Fax: +62-711-355918*

*Email : [manto\\_6611@yahoo.co.id](mailto:manto_6611@yahoo.co.id); [alan\\_tomponu@mail.polisriwijaya.ac.id](mailto:alan_tomponu@mail.polisriwijaya.ac.id); [alan\\_polsri@yahoo.com](mailto:alan_polsri@yahoo.com)*

## ABSTRAK

*Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud pada penelitian ini adalah gambar statis yang berasal sensor vision berupa webcam. Secara matematis, citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris. Pada pengolahan warna gambar, ada bermacam-macam model salah satunya adalah model rgb atau normalisasi RGB. Model pengolahan ini merupakan pengolahan warna dengan menghitung prosentase warna RGB dalam sebuah citra. Dengan menggunakan model ini, sebuah obyek dengan warna tertentu dapat dideteksi dan terbebas dari pengaruh perubahan intensitas cahaya dari luar. Kelemahan dari pengolahan warna model ini adalah tidak dapat membedakan warna hitam dan putih, karena memiliki prosentase nilai RGB yang sama yaitu 33%. Guna melihat pengaruh pendeteksian obyek terhadap perubahan intensitas cahaya maka nilai brightness diubah-ubah. Berdasarkan hasil tersebut pada saat nilai brightness antara 1 – 80 obyek target yang diinginkan masih dapat dideteksi.*

**Kata Kunci :** *Citra Digital, Deteksi Obyek, Normalisasi RGB*

### 1. Pendahuluan.

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer [1].

Secara matematis, citra merupakan fungsi kontinyu (continue) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reperesentasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra.

Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra.

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

Suatu citra  $f(x,y)$  dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

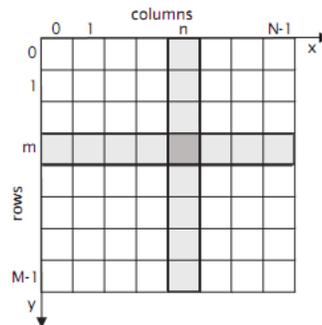
$$0 \leq f(x,y) \leq G-1$$

dimana :  
 $M$  = jumlah piksel baris (*row*) pada array citra  
 $N$  = jumlah piksel kolom (*column*) pada array citra  
 $G$  = nilai skala keabuan (*graylevel*)

Besarnya nilai  $M$ ,  $N$  dan  $G$  pada umumnya merupakan perpangkatan dari dua.

$$M = 2^m; N = 2^n; G = 2^k \quad (2)$$

dimana nilai  $m$ ,  $n$  dan  $k$  adalah bilangan bulat positif. Interval (0,G) disebut skala keabuan (*grayscale*). Besar  $G$  tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai  $G$  sama dengan  $2^8 = 256$  warna (derajat keabuan).



Gambar 1: Representasi citra digital dalam 2 dimensi[3].

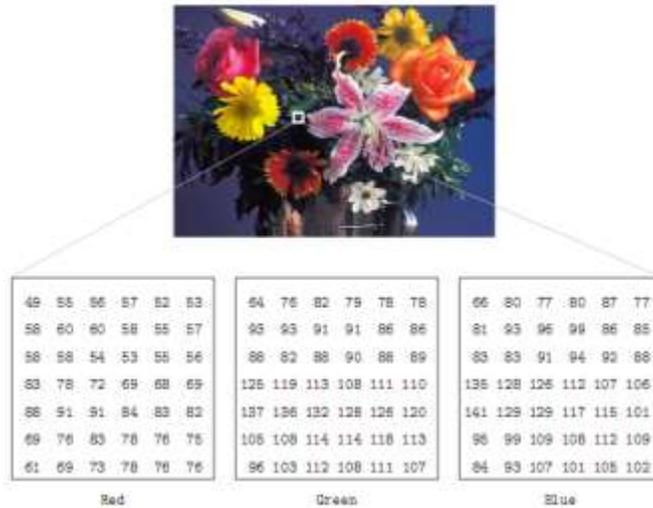
Obyek tertentu dapat dideteksi dengan menggunakan pengolahan citra digital ini. Salah satu metode yang digunakan adalah berdasarkan segmentasi warna. Normalisasi RGB adalah salah satu metode segmentasi warna yang memiliki kelebihan yaitu mudah, proses cepat dan efektif pada obyek *traffic sign* [4], maupun aplikasi untuk *face detection* [5][6].

## 2. Jenis Citra Digital.

Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi 3, *color image*, *black and white image* dan *binary image*.

### 1. *Color Image* atau RGB (*Red, Green, Blue*).

Pada *color image* ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika masing-masing warna memiliki range 0 - 255, maka totalnya adalah  $255^3 = 16.581.375$  (16 K) variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap pixel, gambar tersebut juga disebut gambar-bit warna. *Color image* ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan gambar 2.



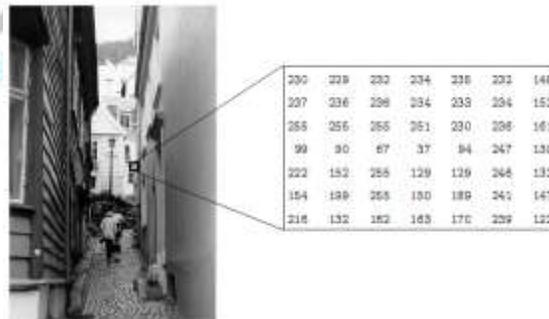
Gambar 2: Color Image[7].

2. *Black and White.*

Citra digital *black and white* (*grayscale*) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 byte. Rentang warna pada *black and white* sangat cocok digunakan untuk pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam kedokteran (X-ray). *Black and white* sebenarnya merupakan hasil rata-rata dari *color image*, dengan demikian maka persamaannya dapat dituliskan sebagai berikut :

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (3)$$

dimana  $I_R(x, y)$  = nilai piksel *Red* titik  $(x, y)$ ,  $I_G(x, y)$  = nilai piksel *Green* titik  $(x, y)$ ,  $I_B(x, y)$  = nilai piksel *Blue* titik  $(x, y)$  sedangkan  $I_{BW}(x, y)$  = nilai piksel *black and white* titik  $(x, y)$ .



Gambar 3: Black and White (Grayscale)[7]

3. *Binary Image.*

Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (*finger print*), atau gambar arsitektur.

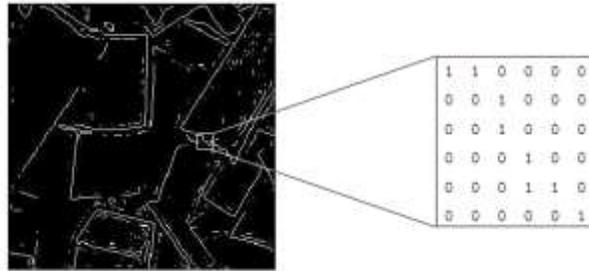
*Binary image* merupakan hasil pengolahan dari *black and white image*, dengan menggunakan fungsi sebagai berikut :

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (4)$$

dan dalam bentuk *floating point*

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (5)$$

dimana  $I_{BW}(x, y)$  = nilai piksel *Gray* titik  $(x, y)$ ,  $I_{Bin}(x, y)$  = nilai piksel *Binary* titik  $(x, y)$ , sedangkan  $T$  adalah nilai *threshold*.



Gambar 4: *Binary Image*[7]

### 3. Segmentasi warna normalisasi RGB.

Segmentasi warna, ada bermacam-macam model warna. Model RGB (*Red Green Blue*) merupakan model yang banyak digunakan, salah satunya adalah monitor. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain model RGB terdapat juga model normalisasi RGB dimana model ini terdapat 3 komponen yaitu,  $r$ ,  $g$ ,  $b$  yang merepresentasikan prosentase dari sebuah piksel pada citra digital [5][8]. Nilai-nilai tersebut mengikuti persamaan-persamaan dibawah ini :

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (6)$$

$$\text{, sehingga : } r + g + b = 1 \quad (7)$$

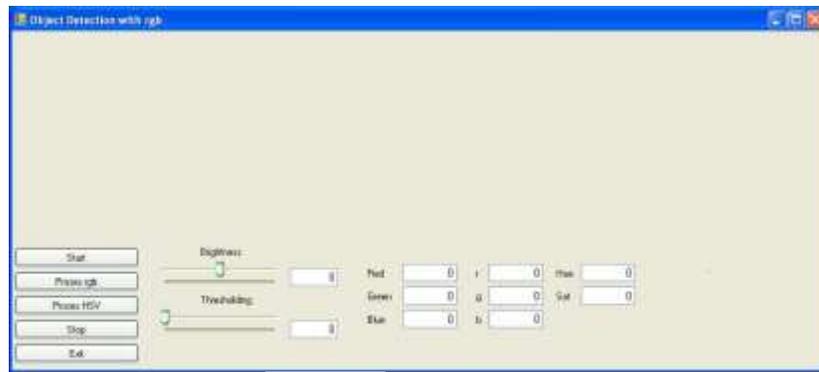
Dengan demikian berdasarkan persamaan 7 maka cukup hanya menggunakan  $r$  dan  $g$  saja, karena nilai  $b$  bisa didapatkan dengan menggunakan  $b = 1 - r - g$ .

### 4. Pengolahan citra menggunakan EmguCV.

EmguCV adalah cross platform yang terdapat dalam .NET untuk *library* pengolahan citra pada Intel OpenCV. EmguCV ini mengikuti fungsi yang terdapat pada OpenCV yang diambil dari .NET oleh sebab itu *compatible* dengan bahasa pemrograman C#, VB, VC++, IronPython dan sebagainya. Program ini bersifat *opensource* sehingga sangat cocok apabila digunakan untuk penelitian, salah satunya adalah untuk aplikasi *computer vision*.

### 5. Hasil dan pembahasan.

Pada pengujian untuk menentukan klasifikasi warna dengan menggunakan model segmentasi warna normalisasi RGB ini menggunakan program Visual Studio 2008 yang telah dilengkapi dengan program pendukung EmguCV. Obyek yang digunakan adalah berupa bola. Pemilihan obyek ini dikarenakan mulai 2011 ini *event robosoccer* mulai dilombakan secara nasional, dimana bola yang digunakan berwarna *orange* [9].



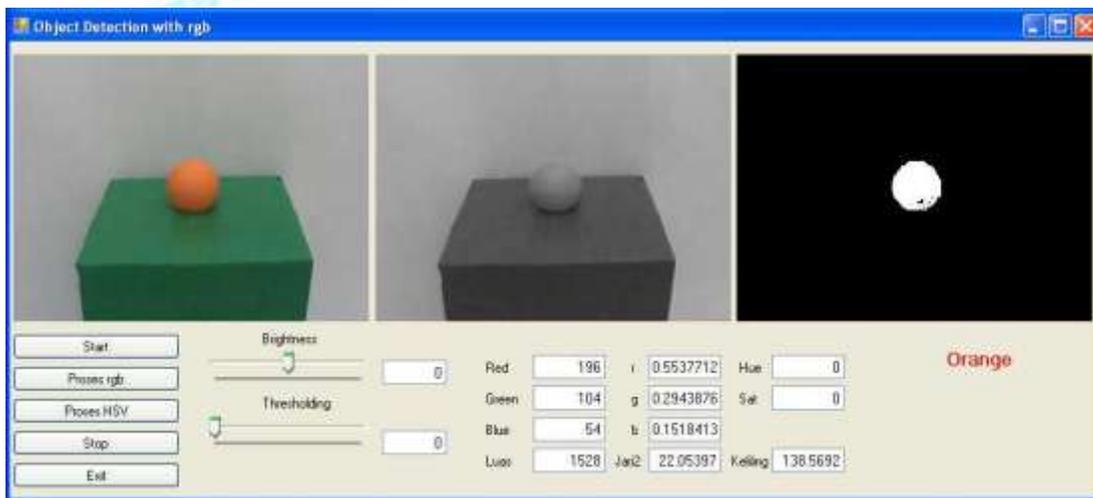
Gambar 5: Program simulasi

Langkah awal untuk membuat segmentasi warna dengan normalisasi RGB ini adalah mengurai data RGB sebanding dengan ukuran pikselnya. Data RGB yang diperoleh kemudian dihitung dengan menggunakan persamaan 6.

```

For i = 0 To ImgColor.Height - 1
    For j = 0 To ImgColor.Width - 1
        BlueData = ImgColor.Data(i, j, 0)
        GreenData = ImgColor.Data(i, j, 1)
        RedData = ImgColor.Data(i, j, 2)
        b = BlueData / (BlueData + GreenData + RedData)
        g = GreenData / (BlueData + GreenData + RedData)
        r = RedData / (BlueData + GreenData + RedData)
        If (r < 0.6 And r > 0.5) And (g < 0.33 And g > 0.28) Then
            ImgGray.Data(i, j, 0) = 255
            areal = areal + 1
        Else
            ImgGray.Data(i, j, 0) = 0
            areal = areal + 0
        End If
    Next
Next
    
```

Berdasarkan pengukuran bahwa warna *orange* ini terletak antara  $r = 0,5 - 0,6$ . Apabila nilai tersebut digunakan sebagai nilai *threshold*, dimana jika diluar nilai itu piksel diberi nilai 0 dan diantara nilai *threshold* diberi nilai = 255 maka obyek bola akan tampak berwarna putih.



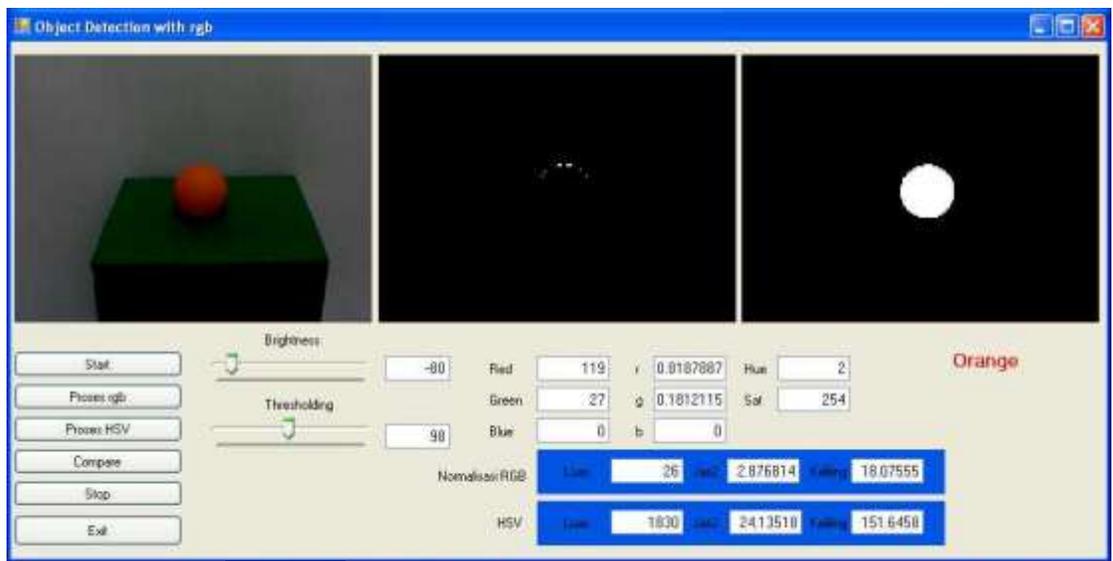
Gambar 6: Hasil pendeteksiian bola menggunakan normalisasi RGB.

Setelah bola berwarna putih sedangkan semua *background* berwarna hitam, maka luas lingkaran dapat diketahui dimana satuannya adalah piksel. Berdasarkan luas lingkaran ini maka parameter untuk jari-jari dan keliling dapat dicari dengan menggunakan persamaan yang terdapat dalam lingkaran.

```

    TextBox11.Text = areal
    jari2 = Math.Sqrt(areal / Math.PI)
    TextBox12.Text = jari2
    keliling = 2 * (Math.PI) * jari2
    TextBox13.Text = keliling
    ImageBox3.Image = ImgGray
    
```

Guna mengetahui pengaruh hasil normalisasi RGB ini terdapat intensitas cahaya, maka digunakan simulasi nilai *brightness* antara -100 sampai 100. Hasil perubahan ini akan dibandingkan dengan segmentasi warna yang lain yaitu HSV. Dengan membandingkan luas lingkaran antara dua model tersebut maka akan diketahui prosentase *error* dari luas yang dihitung.



Gambar 6: Hasil perbandingan normalisasi RGB dengan HSV.  
Tabel 1 : Hasil perbandingan luas akibat perubahan *brightness*

HSV		Normasasi RGB		error (piksel)
Brightness	Luas Lingkaran (piksel)	Brightness	Luas Lingkaran (piksel)	
-100	1837	-100	9	1828
-100	1837	-80	26	1811
-100	1837	-60	35	1802
-100	1837	-40	53	1784
-100	1837	-20	696	1141
-100	1837	0	1632	205
-100	1837	20	1525	312
-100	1837	40	615	1222
-100	1837	60	0	1837
-100	1837	80	0	1837

-100	1837	100	0	1837
------	------	-----	---	------

## 6. Penutup.

### 6.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan dapatkan kesimpulan sebagai berikut :

1. Model normalisasi RGB ini sangat mudah untuk diaplikasikan khususnya untuk mendeteksi obyek dengan warna-warna tertentu.
2. Berdasarkan persamaan 6, warna putih dan hitam sulit dibedakan karena memiliki nilai  $r, g, b$  yang sama untuk kedua warna.
3. Pada saat nilai *brightness* 0 hasil yang dicapai untuk pengenalan obyek berupa bola adalah maksimal.

### 6.2 Rekomendasi.

Berdasarkan penelitian yang dilakukan, penerapan normalisasi RGB ini sangat mungkin apabila diaplikasikan pada *embedded system* mengingat perhitungan yang dilakukan tidak terlalu banyak. Mengingat memory yang terdapat pada *embedded system* tidak teralu besar maka disarankan menggunakan citra digital dengan ukuran piksel kecil. Kedepannya akan dikembangkan untuk mendeteksi warna biru dan kuning sebagai representasi gawang. Model HSV akan digunakan juga dapat diketahui dari kedua model ini yang dapat secara maksimal untuk mendeteksi bola dan gawang.

### Daftar Pustaka

- [1] Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, “ *Teori Pengolahan Citra Digital* ”, Andi Yogyakarta dan UDINUS Semarang, 2009.
- [2] Purnomo Mauridhi Hery, Muntasa Arif, “ *Konsep Pengolahan Citra Digital dan Ekstraksi Fitur* “, Graha Ilmu Yogyakarta, 2010.
- [3] Jähne Bernd, Haußecker Horst, “ *Computer Vision and Applications* “, San Diego, California, Academic Press, 2000.
- [4] Aryuanto, Somawirata Komang, Limpraptono. F. Yudi, “ *A New Color Segmentation Method Based on Normalized RGB Chromaticity Diagram* ” , ISSN 2085 – 973, Seminar on Intelligent Technology and Its Applications, 2009.
- [5] KONG Wan-zeng, ZHU Shan-an, “ *Multi-face detection based on downsampling and modified subtractive clustering for color images* “, Journal of Zhejiang University SCIENCE A, ISSN 1009-3095 (Print); ISSN 1862-1775 (Online), Received Jan. 9, 2006; revision accepted Aug. 2, 2006
- [6] Dimitrova Desislava, Popov Antony, “ *Finding face features in color images using fuzzy hit-or-miss transform* “, 9<sup>th</sup> WSEAS International Conference on FUZZY SYSTEM (FS’08) which was held in Sofia, Bulgaria, 2008
- [7] McAndrew Alasdair, (2004), *An Introduction to Digital Image Processing with Matlab. Notes for SCM2511 Image Processing 1*, School of Computer Science and Mathematics Victoria University of Technology.
- [8] Jähne Bernd, Haußecker Horst, “ *Computer Vision and Applications* “, San Diego, California, Academic Press, 2000.
- [9] Direktorat Penelitian dan Pengabdian kepada Masyarakat, Direktorat Jenderal Pendidikan Tinggi, Kementerian Pendidikan Nasional, ” *Panduan Kontes Robot Cerdas Indonesia (KRCI 2011)*”, 2011.