

SEGMENTASI CITRA UNTUK DETEKSI OBJEK WARNA PADA APLIKASI PENGAMBILAN BENTUK CITRA RECTANGLE

Asep Nana H^[1], M. Ichwan^[1], I Made Santika Putra

^[1]Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional Bandung

asepnana@itenas.ac.id, ichwan@itenas.ac.id, imadesantikaputra@gmail.com

ABSTRAK

Pengambilan citra terkadang tidak sesuai dengan yang dibutuhkan, sehingga diperlukan proses pengolahan citra untuk mendapatkan bentuk citra yang diinginkan. Dengan menanamkan kecerdasan pada sebuah kamera, pengambilan suatu citra dengan bentuk tertentu dapat dilakukan sehingga mengurangi beban waktu dan memfasilitasi pengguna dalam proses pengambilan gambar. Segmentasi citra merupakan proses pengambilan informasi dari citra dalam pencarian citra yang serupa seperti warna. Warna dapat dijadikan input dalam penggambaran daerah yang diinginkan (*Region Of Interest*) melalui proses deteksi warna dan *tracking* warna, sehingga dapat dilakukan pengambilan gambar dalam bentuk tertentu. Object Tracking adalah proses mengikuti suatu objek yang bergerak dan berpindah posisi. Computer Vision didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati. Dalam mengenali obyek yang diamati dilakukan proses segmentasi citra dengan menggunakan euclidean color filtering, grayscale dan deteksi blob. Dalam implementasi segmentasi citra untuk deteksi objek warna untuk pengambilan bentuk citra rectangle, digunakan metode segmentasi dan center of gravity. Hasil dari pengujian sistem ini adalah objek warna dapat terdeteksi dan dijejaki dengan baik dalam jarak terbaik antara 40cm-80cm dan dengan intensitas cahaya terbaik antara 22lx-242lx.

Kata Kunci : *Pengolahan Citra, Computer Vision, Deteksi Warna, Segmentasi, Objek Tracking, Capture image, Center of Gravity.*

ABSTRACT

In order to retrieve an appropriate image, an image processing needed to obtain the desired of image form. Instilling artificial intelegent on a camera, capturing an image with a certain shape can be reducing the load time and facilitates the user in the process of shooting. Image segmentation is the process of retrieval information from seeking a similar image like a color. Color can be used as inputs in the depiction of the desired area (Region Of Interest) through the process of color detection and tracking of color, so it can be shooting in a particular form. Object Tracking is the process of following a moving object and switch positions. Computer Vision is defined as a branch of science that studies how computers can recognized the observed objects. In recognition of the observed object, there is segmentation process using euclidean color filtering, gray scale and blob detection. In the implementation of image segmentation to detect color of the object and to capture the image of the rectangle shape, there is methods of segmentation and center of gravity. Results from the testing of this system is the color object can be detected and traced by the best distance between 40cm-80cm best and with the best light intensity between 22lx-242lx.

Keywords : *Image Processing, Computer Vision, Color Detection, Segmentation, Objek Tracking, Capture image, Center of Gravity.*

Latar Belakang

Webcam merupakan sebuah *device* yang dapat digunakan sebagai sensor dalam mendeteksi sebuah benda bergerak melalui proses pengolahan citra. Webcam juga dapat digunakan dalam pengambilan gambar (*capture image*).

Object Tracking adalah proses mengikuti suatu objek yang bergerak dan berpindah posisi. *Computer Vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati. Dalam proses pengenalan objek dan deteksi objek diperlukan pemisahan segmen tertentu pada suatu citra yang dikenal dengan proses segmentasi.

Segmentasi citra merupakan bagian dari proses pengolahan citra. Kegunaan segmentasi menurut Forsyth dan Ponce (2003) adalah pengambilan informasi dari citra seperti pencarian bagian mesin, pencarian manusia dan pencarian citra yang serupa. Secara umum pendekatan segmentasi citra yang sering digunakan adalah melalui pendekatan intensitas, pendekatan warna dan pendekatan bentuk (Rujikietgumjorn, 2008). Salah satu produk teknologi dalam proses pengolahan citra adalah *Aforge.Net Framework*.

Pada penelitian ini sistem yang dibangun adalah suatu aplikasi yang dapat mengambil sebuah bentuk citra *rectangle* dari hasil *Region Of Interest* (ROI) berdasarkan objek warna yang dideteksi dan dilacak. Pada Penelitian ini digunakan beberapa metode pelacakan (*tracking*) yakni dengan mengkombinasikan metode segmentasi dan *center of gravity* (COG) yang diharapkan dapat melacak pergerakan dari objek warna yang dideteksi.

Rumusan Masalah

Rumusan permasalahan yang ada pada penelitian ini adalah :

1. Bagaimana sistem dapat mendeteksi dan melacak

pergerakan objek warna yang ditempelkan di jari pengguna.

2. Bagaimana sistem dapat menentukan poin 1 dan poin 2 dari objek warna yang bergerak.
3. Bagaimana sistem dapat membentuk garis *rectangular* berdasarkan poin 1 dan poin 2 dari pergerakan objek warna.
4. Bagaimana sistem dapat mengambil bentuk citra berdasarkan pembentukan garis *rectangular*.

Tujuan Penelitian

Tujuan dari kegiatan penelitian ini adalah mendeteksi adanya warna yang bergerak dalam penentuan letak dengan menggunakan metode pelacakan yakni metode segmentasi warna dan *center of gravity* (COG) sehingga dapat menyeleksi suatu citra tertentu berdasarkan daerah yang diinginkan (*Region Of Interest*).

Batasan Masalah

Batasan masalah adalah sebagai berikut :

1. Aplikasi yang dibuat adalah aplikasi berbasis *desktop*.
2. Menggunakan webcam untuk menangkap pergerakan objek warna.
3. Warna yang digunakan pada objek yang di *tracking* adalah warna yang berbeda dengan latar warna *background* dan objek yang diseleksi serta di *capture*.
4. Garis yang terbentuk dari proses *tracking* warna berdasarkan koordinat awal dan koordinat akhir adalah garis dengan bentuk persegi (*rectangular*).
5. Resolusi dari keluaran bentuk citra yang dihasilkan bergantung pada kamera webcam yang digunakan.
6. Metode yang digunakan metode pelacakan yakni mengkombinasikan metode segmentasi warna dan *center of gravity* (COG).
7. Proses segmentasi warna menggunakan framework *aforge.Net*

Framework Aforge.Net ^[6]

AForge.NET merupakan framework C# terbuka yang dirancang untuk pengembang dan peneliti di bidang Computer Vision dan *Artificial Intelligence* (Kecerdasan Buatan) - Pengolahan Gambar, Jaringan Saraf Tiruan, Algoritma Genetika, Logika Fuzzy, Pembelajaran Mesin, Robotika, dll.

Kerangka ini terdiri oleh set perpustakaan dan contoh aplikasi, yang menunjukkan karakteristiknya:

- AForge.Imaging - library dengan rutinitas pengolahan citra dan filter;
- AForge.Vision - library *computer vision*;
- AForge.Video - set library untuk pemrosesan video;
- AForge.Neuro - library perhitungan jaringan saraf tiruan;
- AForge.Genetic - library pemrograman evolusi;
- AForge.Fuzzy - library perhitungan *fuzzy*;
- AForge.Robotics - library yang memberikan dukungan pada beberapa robotika kit;
- AForge.MachineLearning - library pembelajaran mesin; dan lain-lain.

Euclidean Color Filtering

Euclidean Color Filtering adalah metode yang berguna untuk menemukan sebuah warna yang terdapat pada sebuah gambar. Filter ini memfilter piksel-piksel pada gambar yang berada di dalam/di luar dari lingkup RGB (Red Green Blue) dengan pusat dan radius tertentu. Filter tersebut membiarkan piksel-piksel dengan warna yang berada di dalam/di luar dari lingkup yang telah ditentukan dan mengisi sisanya dengan warna tertentu.

Grayscale

GrayScale adalah suatu citra dimana nilai dari setiap pixel merupakan sampel tunggal. Citra yang ditampilkan adalah citra keabuan dimana intensitasnya berada pada interval 0 – 255, warna hitam (0) pada bagian yang intensitasnya

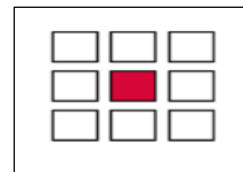
terlemah dan warna putih(255) pada intensitas terkuat. Proses konversi dari citra berwarna menjadi Grayscale menggunakan koefisien dari ITU Recommendation BT.709.

$$\text{Grayscale} = 0.2125 \cdot \text{red} + 0.7154 \cdot \text{green} + 0.0721 \cdot \text{blue}.$$

Persamaan Konversi Grayscale(1)

Blob ^[4]

Blob merupakan sekumpulan piksel-piksel yang memiliki hubungan tetangga. Proses perhitungan *blob* dapat dilakukan dengan melakukan analisis piksel yang bertetangga. Piksel bertetangga pada sebuah piksel ditentukan sebagai piksel yang berjarak 1 dari piksel asal. Proses perhitungan *blob* akan memanfaatkan relasi piksel *8-neighbors*. Gambar 1 merupakan gambaran sederhana dari relasi *8-neighbors*.



Gambar 1. Relasi 8-neighbors

P(x-1, y-1)	P(x, y-1)	P(x+1, y-1)
P(x-1, y)	P(x, y)	P(x+1, y)
P(x-1, y+1)	P(x, y+1)	P(x+1, y+1)

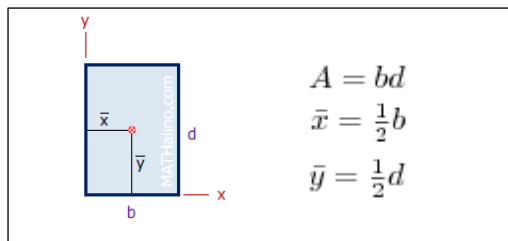
Relasi 8-neighbors

Proses pemetaan objek akan menelusuri tiap piksel pada setiap baris yang ada dan memberikan label pada piksel yang memiliki nilai warna selain hitam (RGB = 0 0 0). Setiap piksel yang memiliki relasi hubungan *8-neighbors* akan diberikan label yang sama.

Pusat Massa Obyek (*Center Of Gravity*)^[9]

Metode Center of Gravity adalah metode yang dipergunakan untuk menentukan titik keseimbangan dari grafik yang merupakan hasil dari proses pengolahan citra. Persamaan 3

merupakan perumusan matematis dari metode center of gravity.



Persamaan COG (3)

Analisis Sistem

Kebutuhan sistem aplikasi deteksi objek warna untuk pengambilan bentuk citra berupa kebutuhan akan *hardware* dan *software*.

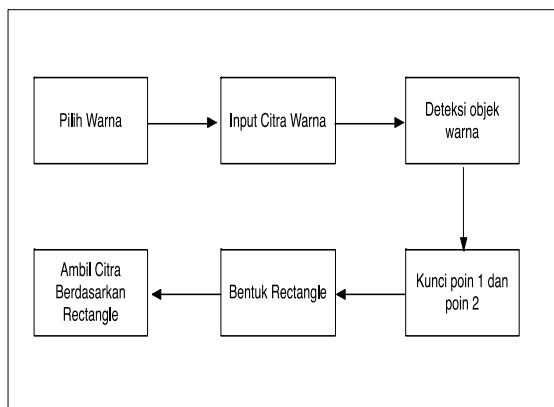
- Analisis fitur sistem

Dalam perancangan aplikasi deteksi objek warna untuk pengambilan bentuk citra, disusun sebuah analisis terhadap fitur yang akan diterapkan, seperti :

- Mendeteksi dan menjejaki objek warna.
- Menampilkan garis *rectangle* untuk pengambilan bentuk citra dari proses penguncuan poin 1 dan poin 2.

- Blok Diagram

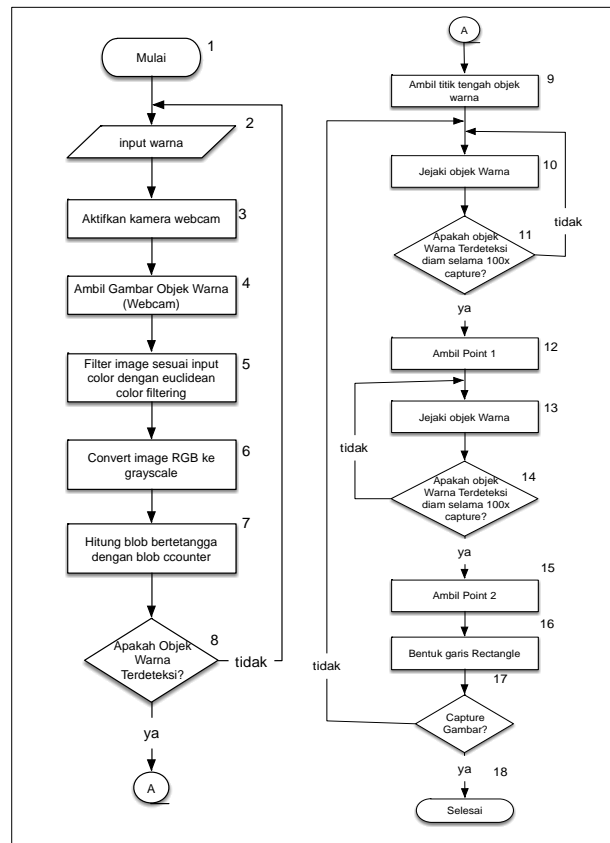
Gambar 2 adalah blok diagram atau alur kerja dari aplikasi deteksi objek warna untuk pengambilan bentuk citra.



Gambar 2. Blok Diagram

Sistem Kerja Aplikasi

Flowchart atau diagram alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Gambar 3 adalah perancangan flowchart aplikasi deteksi objek bergerak untuk pengambilan bentuk citra.



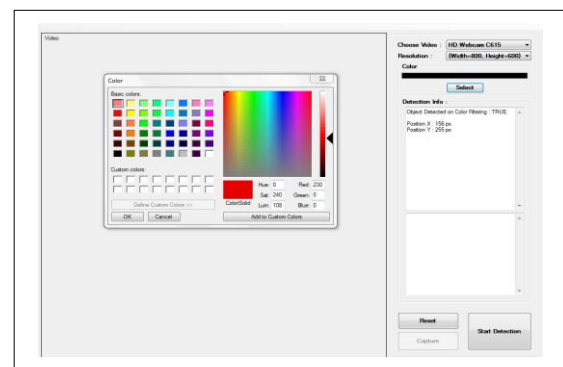
Gambar 3. FlowChart Sistem

Berdasarkan sistem kerja aplikasi seperti terlihat pada gambar 3, terdapat beberapa tahapan dalam implementasi segmentasi citra untuk deteksi objek warna pada pengambilan citra *rectangle*, berikut adalah tahapan-tahapan dari gambar 3 :

1. Mulai

Merupakan *state* awal memulai aplikasi.

2. Pemilihan Warna



Gambar 4. *Pallate* Warna untuk pemilihan warna

Pada tahap pemilihan warna dilakukan sebagai masukan kepada sistem

untuk mendeteksi warna yang akan dideteksi dan dijejaki. Proses pemilihan warna dapat dilihat pada gambar 4.

3. Pengaktifan kamera webcam

Pada tahap proses pengaktifan perangkat kamera webcam dilakukan sebagai media untuk mendeteksi objek. Pengaktifan kamera webcam dilakukan dengan menekan tombol *start detection* pada aplikasi. Pada gambar 5 merupakan kondisi ketika media webcam aktif.



Gambar 5. Kamera Webcam Aktif

4. Input Objek Warna

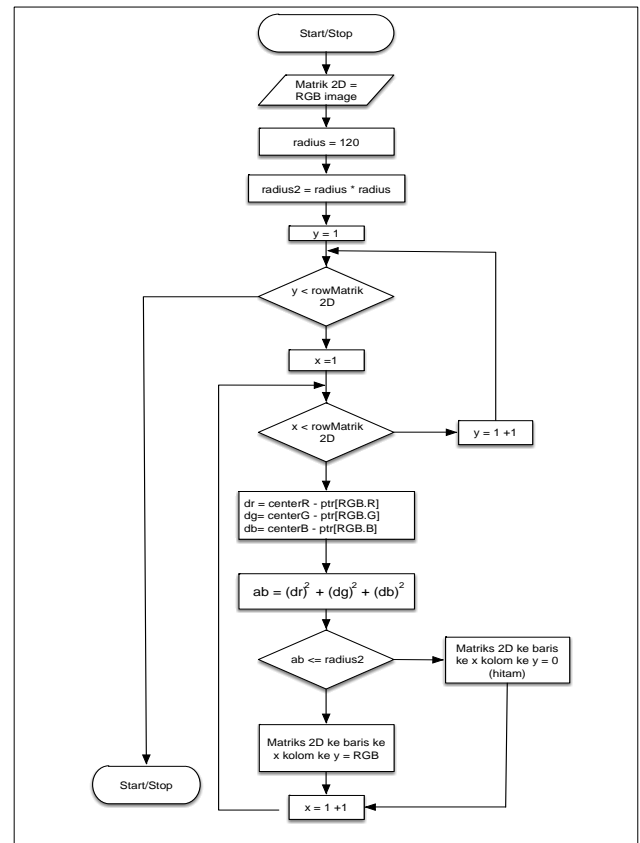
Pada tahap ini dilakukan input objek warna dari warna yang sebelumnya dipilih pada tahap 1. Gambar 6 adalah proses input objek warna.



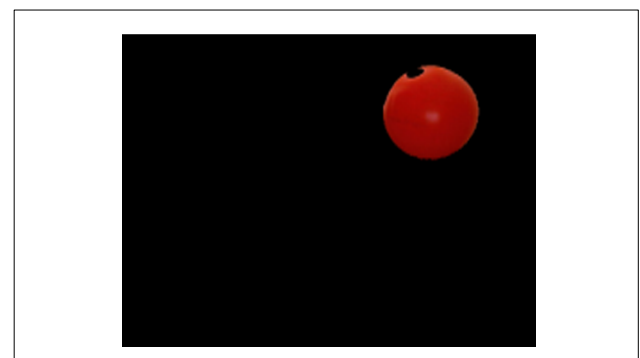
Gambar 6. Input Objek Warna

5. Segmentasi objek warna terhadap ruang warna RGB dengan Euclidean Color Filtering

Pada tahap ini dilakukan proses *filtering* warna menggunakan *euclidean color filtering* berdasarkan pemilihan warna pada tahap 1. Proses *filtering* menggunakan *euclidean color filtering* ini adalah memisahkan objek warna dari background dimana background akan dijadikan warna hitam seperti terlihat pada gambar 8. Proses euclidean dapat dilihat pada gambar 7.



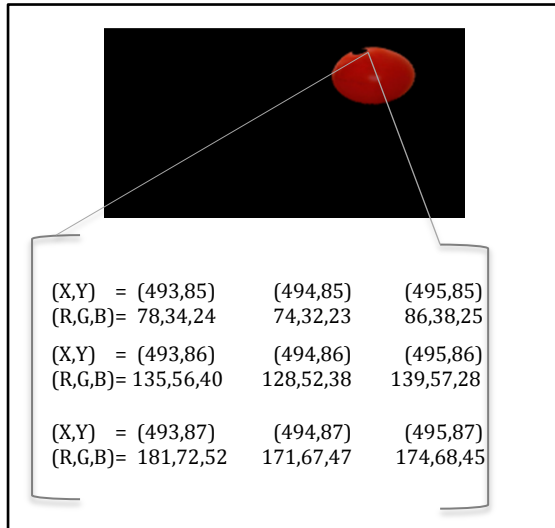
Gambar 7. Flowchart Proses Euclidean Color Filtering



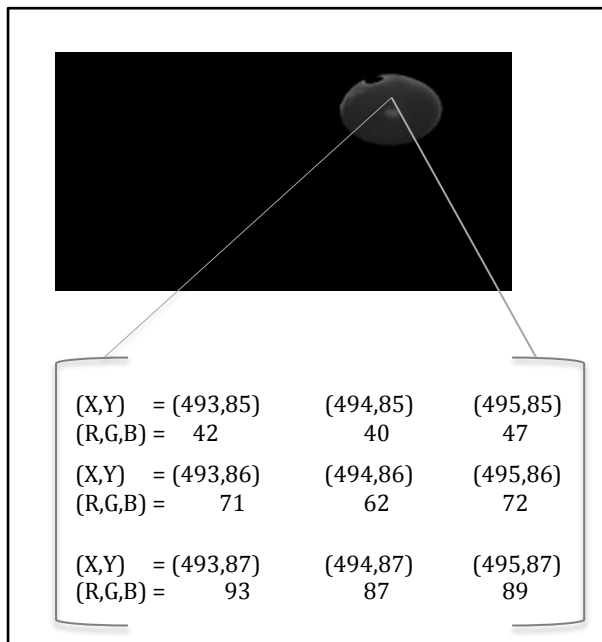
Gambar 8. Input citra setelah dilakukan euclidean color filtering

6. Konversi Warna RGB ke Grayscale

Setelah dilakukan pemisahan objek warna dengan menggunakan *euclidean color filtering*, *state* selanjutnya adalah mengubah nilai warna citra rgb ke dalam ruang warna *grayscale* (keabuan). Proses mengubah nilai warna citra rgb ke dalam ruang warna *grayscale* menggunakan persamaan 1. ilustrasi perubahan nilai warna citra rgb dapat dilihat pada gambar 9 dan gambar 10.



Gambar 9. Ilustrasi RGB pada piksel citra sebelum dikonversi



Gambar 10. Ilustrasi RGB pada piksel citra setelah dikonversi

Berikut contoh proses perhitungan nilai rata-rata citra pada titik koordinat (493,85), (494,86), (495,87) dengan persamaan 1.

a. Koordinat (493,85)

$$\begin{aligned} \text{Grayscale} &= (0.2125 \times R) + (0.7154 \times G) + (0.0721 \times B) \\ &= (0.2125 \times 78) + (0.7154 \times 34) + (0.0721 \times 24) \\ &= 16.575 + 24,3236 + 1,7304 \\ &= 42,629 = 42 \end{aligned}$$

b. Koordinat (494,86)

$$\begin{aligned} \text{Grayscale} &= (0.2125 \times R) + (0.7154 \times G) + (0.0721 \times B) \\ &= (0.2125 \times 128) + (0.7154 \times 52) + (0.0721 \times 38) \\ &= 27,2 + 37,2008 + 2,7398 \\ &= 62,1406 = 62 \end{aligned}$$

c. Koordinat (495,87)

$$\begin{aligned} \text{Grayscale} &= (0.2125 \times R) + (0.7154 \times G) + (0.0721 \times B) \\ &= (0.2125 \times 174) + (0.7154 \times 68) + (0.0721 \times 45) \\ &= 36,975 + 48,6472 + 3,2445 \\ &= 88,867 = 89 \end{aligned}$$

7. Pemetaan dan Perhitungan Blob dengan BlobCounter

Proses perhitungan *blob* akan memanfaatkan relasi piksel *8-neighbors*. Menurut pustaka AForge.Net langkah-langkah perhitungan blob adalah sebagai berikut.

- Proses pemetaan objek akan menelusuri tiap piksel pada setiap baris yang ada dan memberikan label pada piksel yang memiliki nilai warna selain hitam (RGB = 0 0 0). Setiap piksel yang memiliki relasi hubungan *8-neighbors* akan diberikan label yang sama.
- Proses pengumpulan informasi *blob*, akan mengumpulkan dan mengolah informasi tiap piksel yang bertetangga berdasarkan letak dan label yang dihasilkan oleh proses pemetaan objek. Letak dan label piksel yang bertetangga tersebut digunakan untuk membentuk suatu *blob* dan informasi pendukungnya seperti luas area, tingkat kepenuhan, titik pusat dan area kotak *blob*.

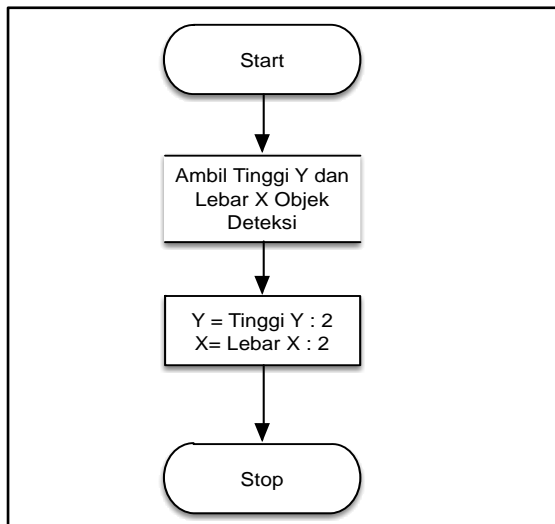
Gambar 11 adalah hasil dari pemetaan dan perhitungan *blob*.



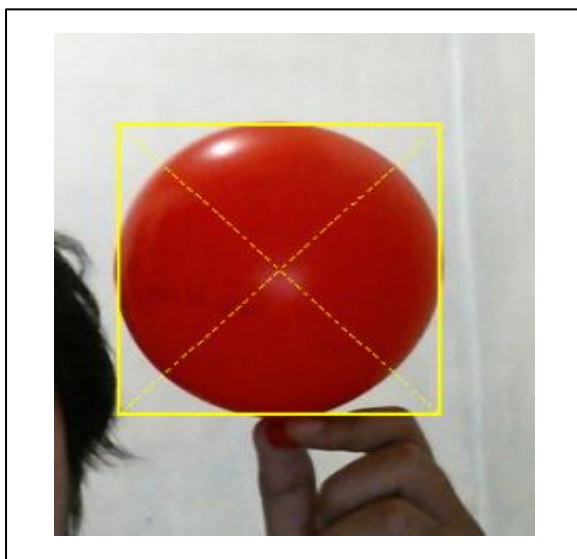
Gambar 11. hasil deteksi blob pada ruang warna RGB

8. Penentuan Titik Pusat (*Center of Gravity*)

Setelah objek warna terdeteksi melalui proses segmentasi warna, grayscale dan blob, maka proses selanjutnya adalah penentuan titik pusat (*center of gravity*) dari objek yang telah dideteksi. Penentuan titik pusat ini nantinya dapat digunakan dalam pembentukan garis *rectangle* dalam proses pengambilan bentuk citra. Proses penentuan titik pusat ini menggunakan persamaan 3 seperti terlihat pada gambar 12. Realisasi dari perhitungan persamaan 3 terlihat pada gambar 13.



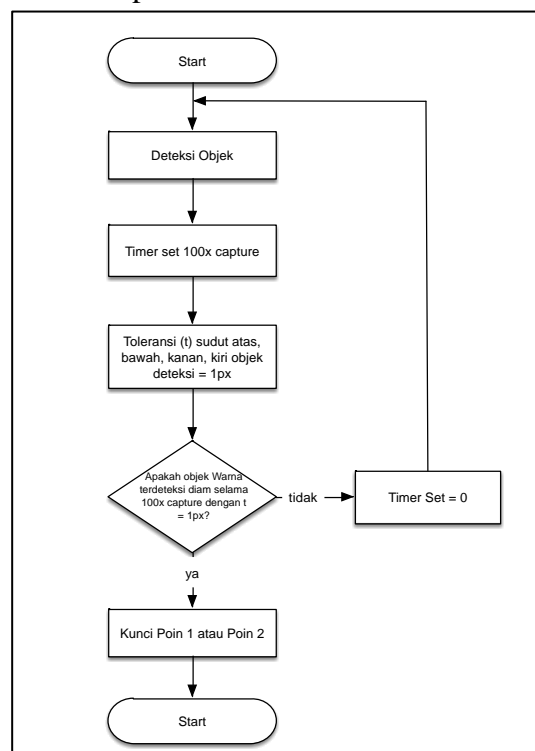
Gambar 12. Flowchart penentuan titik pusat



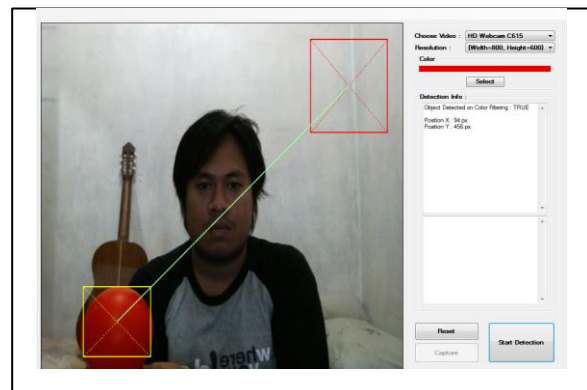
Gambar 13. Titik pusat dari deteksi objek warna

9. Penguncian Poin 1 dan Poin 2

Dalam proses pengambilan bentuk citra *rectangle* dibutuhkan penentuan poin 1 dan poin 2 oleh user. Setelah objek terdeteksi dan telah ditentukan titik pusat, proses selanjutnya adalah penguncian poin 1 dan poin 2. Poin 1 dan poin 2 dapat terkunci jika objek yang terdeteksi diam selama 100x capture dalam toleransi 1px atas, 1px bawah, 1px kanan dan 1px kiri. Gambar 14 adalah *flowchart* proses penguncian poin 1 dan poin 2. Gambar 15 adalah realisasi dari *flowchart* penguncian poin 1 dan poin 2.



Gambar 14. Flowchart Penguncian Poin 1 dan poin 2



Gambar 15. Realisasi penguncian poin 1 dan poin 2

10. Pembentukan *rectangle*

Setelah penguncian poin 1 dan poin 2, maka sistem secara otomatis akan membentuk garis *rectangle*. Proses pembentukan *rectangle* seperti terlihat pada Gambar 16. Realisasi proses pembentukan *rectangle* seperti terlihat pada Gambar 17.

```
private void lock_rectangle(Graphics g, Bitmap mImage, Bitmap realImage)
{
    using (Pen pen = new Pen(Color.FromArgb(160, 255, 160), 2))
    {
        //g.DrawRectangle(pen, objectRect);
        if (endPoint == true && posXA != 0 && posXB != 0)
        {
            int posXAA = 0;
            int posYAA = 0;
            int posXBB = 0;
            int posYBB = 0;

            if (posXB - posXA < 0)
            {
                posXAA = posXB + (widthB/2);
                posXBB = posXA + (widthA/2);
            }
            else
            {
                posXAA = posXA + (widthA/2);
                posXBB = posXB + (widthB/2);
            }

            if (posYB - posYA < 0)
            {
                posYAA = posYB + (heightB/2);
                posYBB = posYA + (heightA/2);
            }
            else
            {
                posYAA = posYA + (heightA/2);
                posYBB = posYB + (heightB/2);
            }

            pointRect = Rectangle.FromLTRB(posXAA, posYAA, posXBB, posYBB);
            g.DrawRectangle(pen, pointRect);
        }
    }
}
```

Gambar 16. Proses Pembentukan *Rectangle*



Gambar 17. Realisasi Pembentukan *Rectangle*

11. Pengambilan Citra *Rectangle*

Setelah pembentukan garis *rectangle* maka untuk pengambilan bentuk citra *rectangle* dilakukan penekanan pada tombol *capture* yang terdapat pada aplikasi. Hasil proses *capture* seperti terlihat pada Gambar 18. Format gambar dari hasil pengambilan citra berdasarkan pembentukan garis *rectangle* adalah berformat .jpg.



Gambar 18. Hasil Pengambilan gambar

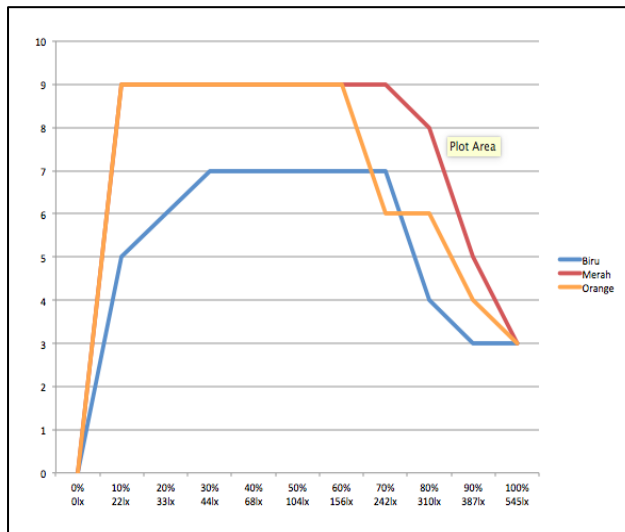
Pengujian Deteksi Terhadap Cahaya dan Jarak

Pengujian pada aplikasi deteksi objek bergerak untuk pengambilan bentuk citra ini meliputi pengujian deteksi objek warna terhadap cahaya, pengujian deteksi objek warna terhadap jarak, dan pengujian blackbox fitur sistem. Hasil pengujian deteksi objek warna terhadap cahaya dan jarak terlihat pada tabel 1.

Tabel 1 tingkat keberhasilan deteksi objek warna terhadap cahaya

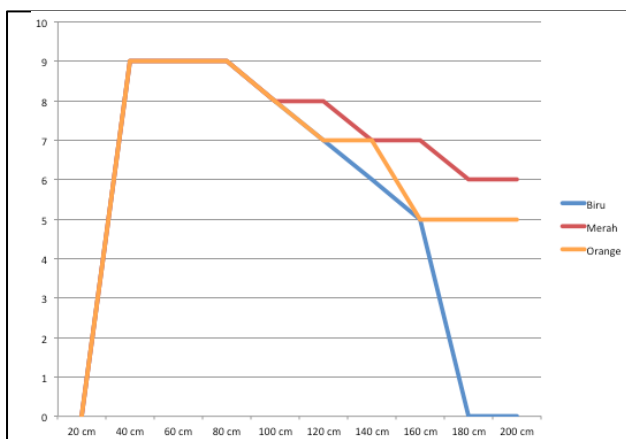
Jarak	Pengujian Deteksi Objek Warna Merah					Nilai Cahaya																
	20	40	60	80	100	120	140	160	180	200	0% = 0 lx	10% = 22 lx	20% = 33 lx	30% = 44 lx	40% = 68 lx	50% = 104 lx	60% = 156 lx	70% = 242 lx	80% = 310 lx	90% = 387 lx	100% = 545 lx	
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
60	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
80	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
100	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
120	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
140	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
160	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
180	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
200	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Pada gambar 19 merupakan grafik pengujian deteksi objek warna terhadap cahaya dari 3 objek warna yang diuji yaitu merah, biru dan orange



Gambar 19. Grafik deteksi warna terhadap intensitas cahaya

Pada gambar 20 merupakan grafik pengujian deteksi objek warna terhadap jarak dari 3 objek warna yang diuji yaitu merah, biru dan orange



Gambar 20. Grafik deteksi warna terhadap jarak

Pengujian Blackbox Fitur Sistem

Pengujian fitur sistem ini dilakukan dengan menggunakan metode *blackbox* agar dapat diketahui fitur sistem dapat berfungsi dengan baik seperti terlihat pada tabel 2. Pengujian fitur sistem yang dilakukan adalah:

1. Pemilihan warna

2. Deteksi objek warna
3. Penguncian poin
4. Pembentukan *Rectangle*
5. Pengambilan Gambar

Tabel 2 Pengujian *Blackbox* fitur sistem

NO	Hasil Pengujian	Keterangan
1		Media pilihan warna muncul/berhasil
2		Deteksi objek warna berhasil
3		Penguncian poin 1 dan poin 2 berhasil
4		Pembentukan garis <i>rectangle</i> berhasil
5		Pengambilan gambar bentuk <i>rectangle</i> berhasil

Kesimpulan

Berdasarkan penelitian yang telah dilakukan, didapat kesimpulan bahwa :

1. Segmentasi Citra dan pengambilan titik pusat untuk deteksi objek bergerak dapat diimplementasikan pada aplikasi deteksi objek warna untuk pengambilan citra bentuk *rectangle*.
2. Dari hasil pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa dari proses segmentasi citra untuk deteksi objek warna diketahui jarak terbaik

adalah antara 40cm – 80cm. Sedangkan untuk deteksi warna terhadap intensitas cahaya, nilai lumiance terbaik adalah antara 22lx – 242lx.

3. Dari hasil pengujian pada aplikasi deteksi objek warna untuk pengambilan bentuk citra *rectangle*, didapatkan bahwa warna pada objek warna yang dideteksi tidak boleh sama atau terdapat pada latar *background* atau objek yang akan di *capture*.
4. Dari percobaan 3 warna yang dideteksi, warna merah adalah warna yang dapat terdeteksi dengan baik.

Daftar Pustaka

- [1] Rahmadi Kurnia. 2009. “*Penjejak Target Benda Pada Gerakan Liner Berdasarkan Warna*”. Padang : Universitas Andalas Padang.
- [2] Wawan Kurniawan. 2011. “*Pengenalan Bahasa Isyarat Dengan Menggunakan Metode Segmentasi Warna Kulit dan Center Of Gravity*”. Jambi : Universitas Jambi.
- [3] Shanti Anggriani Tambunan. 2010. “*Pengembangan Kerangka Prototype Detektor Api Pada Citra Digital Dengan Menerapkan Metode Segmentasi SRM (Statistical Region Merging)*”. Bandung : Institut Teknologi Nasional Bandung.
- [4] Benedictus Yoga Budi Putranto. 2010. “*Segmentasi Warna Citra Dengan Deteksi HSV Untuk Mendeteksi Objek*”. Yogyakarta : Universitas Kristen Duta Wacana.
- [5] Mohamad Ihsan Nurdin. 2013. “*Perbandingan Aplikasi Pengenalan Gender Berdasarkan Citra Wajah Dengan Implementasi Library Shore Terhadap Aplikasi EyeFace Recognition*”. Bandung : Institut Teknologi Nasional Bandung
- [6] Framework aforge.Net. <http://www.aforgenet.com>. Diakses online pada tanggal 1 juli 2014.
- [7] <http://lib.ui.ac.id/file?file=digital/134242-T%2027914-Penerapan%20algoritma-Literatur.pdf>. Diakses pada 18 Agustus 2014
- [8] <http://repository.usu.ac.id/bitstream/123456789/29363/4/Chapter%20II.pdf>. Diakses pada 18 September 2014
- [9] <http://www.mathalino.com/reviewer/engineering-mechanics/centroids-and-centers-gravity>. Diakses online pada 22 September 2014.
- [10] http://id.wikipedia.org/wiki/Microsoft_Visual_Studio. Diakses pada 20 Oktober 2014.
- [11] <http://thesis.binus.ac.id/doc/Bab2/2011-1-00289-if%202.pdf>. Diakses pada 20 November 2014
- [12] [http://msdn.microsoft.com/en-us/library/system.drawing.rectangle.fromlrb\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.drawing.rectangle.fromlrb(v=vs.110).aspx). Diakses online pada 22 November 2014