



Implementation of a Ball Tracking System In Image Processing Using the Find Contour Method

Implementasi Sistem Tracking Bola Pada Pengolahan Citra Menggunakan Metode Find Contour

Ari Muhammad Rizal¹⁾, Muhammad Taufiqurrohman²⁾, Joko Subur³⁾

^{1,2,3)} Electrical Engineering, Hang Tuah University Surabaya, Surabaya, Indonesia

¹⁾ arimrizal94@gmail.com

²⁾ taufiqurrohman@hangtuah.ac.id

³⁾ joko.subur@hangtuah.ac.id

Abstract. One technology that is experiencing rapid development is digital image processing. Digital image processing is a scientific discipline that studies image processing techniques. The use of image management is increasing and is used in various fields, one of them in the field of robotics can be applied as an object detector. Image management is chosen in terms of detection because it is able to recognize objects and colors quickly. In the field of robotics these advantages are used to control robots to navigate, one example is in soccer robots. Soccer robots must be able to detect the presence of the ball to find and carry the ball and kick it towards the goal. The ball in the match system has been determined with a definite color, in this case it is orange. Therefore, the robot must be able to detect the orange ball to detect it. In detecting the ball, the parameters produced in this system are the distance of the robot to the ball, the angle of direction of the ball and also the ball according to the color or not. To solve this problem in this study using a method that is processing the RGB image produced by the camera into HSV after which color filtering is done and then finished using the find contour method. The results of this study noted the accuracy of the system in tracking ball objects by 92.87 percent at the distance of the ball to the camera and 88.19 percent at the angle of direction of the ball to the ball. The level of accuracy is due to the position of the camera when shooting.

Keywords: Image Processing; Balls; Orange; Color Filtering; Find Contour.

Abstrak. Salah satu teknologi yang mengalami perkembangan yang begitu pesat adalah pengolahan citra digital. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Penggunaan pengolahan citra semakin meningkat dan digunakan diberbagai bidang salah satunya pada bidang robotika dapat diaplikasikan sebagai pendeteksi objek. Pengolahan citra dipilih dalam hal pendeteksian karena mampu mengenali objek dan warna secara cepat. Pada bidang robotika kelebihan tersebut digunakan untuk kendali robot bernavigasi, salah satu contohnya adalah pada robot sepak bola. Robot sepak bola harus mampu mendeteksi keberadaan bola untuk mencari dan membawa bola serta menendangnya kearah gawang. Bola pada sistem pertandingan sudah ditentukan dengan warna yang pasti, dalam hal ini adalah berwarna orange. Oleh karena itu, robot harus mampu mendeteksi bola yang berwarna *orange* tersebut untuk mendeteksinya. Dalam mendeteksi bola tersebut parameter yang dihasilkan dalam sistem ini yaitu jarak robot terhadap bola, sudut arah bola dan juga bola tersebut sesuai dengan warnanya atau tidak. Untuk menyelesaikan permasalahan itu pada penelitian ini menggunakan cara yaitu mengolah gambar RGB yang dihasilkan oleh kamera menjadi HSV setelah itu dilakukan color filtering dan selanjutnya diakhir menggunakan metode *findcontour*. Hasil dari penelitian ini dicatat keakuratan sistem dalam *tracking object* bola sebesar 92.87 persen pada jarak bola terhadap kamera dan 88.19 persen pada sudut arah posisi bola terhadap bola. Tingkat keakuratan tersebut karena dipengaruhi posisi kamera dalam pengambilan gambar.

Kata Kunci: Pengolahan Citra; Bola; Warna Orange; Color Filtering; Find Contour.

PENDAHULUAN

Mata adalah indra terbaik yang dimiliki oleh manusia sehingga citra (gambar) memegang peranan penting dalam perspektif manusia. Namun, mata manusia memiliki keterbatasan dalam menangkap sinyal elektromagnetik. Komputer atau mesin pencitraan lainnya dapat menangkap hampir keseluruhan sinyal elektromagnetik mulai dari gamma hingga gelombang radio. Mesin pencitraan dapat bekerja dengan citra dari sumber yang tidak sesuai, tidak cocok, atau tidak dapat ditangkap dengan penglihatan manusia. Hal inilah yang menyebabkan pengolahan citra digital memiliki kegunaan dan spektrum aplikasi yang sangat luas.

Penggunaan pengolahan citra semakin meningkat dan digunakan diberbagai bidang salah satunya pada bidang robotika dapat diaplikasikan sebagai pendeteksi objek. Pengolahan citra dipilih dalam hal pendeteksian karena mampu mengenali objek dan warna secara cepat. Pada bidang robotika kelebihan tersebut digunakan untuk kendali robot bernavigasi, salah satu contohnya adalah pada robot sepak bola.

Pada penelitian sebelumnya pendeteksian objek bola hanya menggunakan *color filtering* HSV tanpa menggunakan metode pengenalan bentuk objek, maka warna yang sama dengan objek bola akan terdeteksi oleh *system*.

Oleh sebab itu pada penelitian ini menerapkan *color filtering* yang lebih akurat dalam mendeteksi bola berwarna dengan menggunakan ruang warna *Hue Saturation Value* (HSV) berfungsi untuk melakukan teknik segmentasi warna yang mampu menoleransi perubahan intensitas cahaya dalam mendeteksi warna benda dan juga membantu mempermudah dalam teknik *object recognition* dengan menggunakan metode *find contour* untuk mendeteksi benda dengan bentuk lingkaran [1].

METODE

Langkah-langkah untuk pendeteksian objek bola dalam hal ini ditunjukkan pada Figure 1:

- Mengambil gambar objek bola menggunakan webcam
- Ekstraksi warna gambar untuk proses filtrasi warna objek dengan background
- Proses morfologi untuk mengurangi noise setelah dilakukan filtrasi warna
- Pengenalan objek menggunakan metode *find contour*

[Figure 1 about here.]

A. RGB To HSV

Suatu proses perubahan dari citra *RGB* ke *HSV*. *HSV* adalah representasi alternatif dari model warna *RGB*. *HSV* merupakan ruang warna yang sangat cocok untuk mengidentifikasi warna-

warna dasar, dimana warna dasar ini digunakan dalam penelitian sebagai warna identifikasi *object* Selain itu, *HSV* menoleransi terhadap perubahan intensitas cahaya. Inilah yang menjadi keunggulan *HSV* dibandingkan dengan ruang warna lainnya. Untuk mempermudah dalam sistem pemetaan warna, maka dipilihlah menggunakan metode filter HSV seperti Figure 2.

[Figure 2 about here.]

Proses konversi warna dari RGB ke HSV digunakan fungsi perintah *cvtColor* dari OpenCV. Perintah lengkap adalah sebagai berikut: *cvtColor(src, dst, COLOR_BGR2HSV)*;

Parameters:

- *src* - input gambar: 8-bit unsigned, 16-bit unsigned (CV_16UC ...), atau floating-point presisi tunggal.
- *dst* - menampilkan gambar dengan ukuran dan kedalaman yang sama dengan *src*.
- *COLOR_BGR2HSV* - Konversi dari BGR ke ruang warna HSV.

B. HSV Filter

HSV filter atau bisa disebut dengan *color filtering* adalah suatu teknik pengolahan citra yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik. HSV filter digunakan untuk mendapat gambar biner, gambar biner hanya memiliki dua warna yaitu putih dan hitam atau hanya memiliki nilai 1 dan 0, nilai 1 untuk warna putih sedangkan 0 untuk warna hitam. Warna yang dipilih pada proses HSV filter akan berwarna putih sedangkan yang tidak dipilih akan diabaikan menjadi warna hitam. Untuk menentukan warna yang dipilih pada proses HSV filter maka pengguna harus menentukan batas filter maksimum dan minimum, dimana pada batas filter minimum dan maksimum perlu dilakukan pengaturan untuk tiap parameter (*hue*, *saturation*, *value*) jadi batas filter HSV ada 6 variabel yaitu:

1. Hue batas minimal.
2. Hue batas maksimal.
3. Saturation batas minimal.
4. Saturation batas maksimal.
5. Value batas minimal.
6. Value batas maksimal.

Batas filter ini merupakan faktor yang mempengaruhi hasil segmentasi citra agar mendapatkan gambar biner yang sesuai dengan warna yang diinginkan oleh pengguna. Namun dari hasil dari HSV filter masih terdapat noise, maka diperlukan sebuah teknik morfologi untuk mengurangi noise pada gambar biner. Terdapat beberapa operasi dasar morfologi yaitu operasi erosi, dan dilasi seperti Figure 3.

[Figure 3 about here.]

Proses konversi warna dari RGB ke HSV digunakan fungsi perintah `inRange` dari `OpenCV`. Perintah lengkap adalah sebagai berikut: `inRange(src, Scalar(Hue LOW, Saturation LOW, Value_LOW), Scalar(Hue HIGH, Saturation HIGH, Value_HIGH), dst);`

Parameters:

- `src` - input gambar: 8-bit unsigned, 16-bit unsigned (`CV_16UC ...`), atau floating-point presisi tunggal.
- Hue LOW – variable untuk nilai Hue batas minimal.
- Saturation LOW – variable untuk nilai Saturation batas minimal.
- Value LOW – variable untuk nilai Value batas minimal.
- Hue HIGH – variable untuk nilai Hue batas maksimal.
- Saturation HIGH – variable untuk nilai Saturation batas maksimal.
- Value HIGH – variable untuk nilai Value batas maksimal.
- `dst` - menampilkan gambar dengan ukuran dan kedalaman yang sama dengan `src`.

C. Erosi (Erosion)

Erosi adalah salah satu dari dua operasi mendasar dalam pemrosesan gambar morfologis yang menjadi dasar semua operasi morfologis lainnya, Erosi atau pengikisan adalah teknik yang bertujuan untuk memperkecil atau mengikis tepi objek yang berarti merubah titik objek (1) yang bertetangga dengan titik latar (0) menjadi titik latar (0), maksudnya erotion ini digunakan untuk menghapus noise putih kecil dari gambar biner yang dihasilkan dari proses HSV filtering seperti yang dapat dilihat pada Figure 4. Dengan proses erosi berakibat penyusutan ukuran obyek image sehingga dapat digunakan untuk memisahkan objek yang saling gandeng satu sama lain. Setelah dilakukan pengikisan pada area titik putih perlu diperjelas lagi gambarnya, karena area putih pada gambar yang diinginkan (warna bola) juga ikut terkikis. maka dilakukanlah teknik dilasi.

[Figure 4 about here.]

Untuk syntax sebagai berikut:

```
Mat erodeElement = cv::getStructuringElement(MORPH_ELLIPSE, Size( 2 * nilaierotionbola + 1, 2 * nilaierotionbola + 1), Point( nilaierotionbola, nilaierotionbola ));
```

```
Ptr <cuda::Filter> erodeFilter = cuda::createMorphologyFilter(MORPH_ERODE, d.thresh.type(), erodeElement);
erodeFilter->apply(d.thresh, d.thresh);
```

D. Dilasi (Dilation)

Dilation adalah kebalikan dari proses erosi, di mana teknik

ini untuk memperbesar segmen objek (citra biner) dengan menambah lapisan disekeliling objek. Atau dengan menjadi titik latar (0) yang bertetangga dengan titik objek (1) menjadi titik objek (1) seperti yang dapat dilihat pada Figure 5. Dalam penelitian ini proses dilasi digunakan untuk memperbesar kembali titik putih pada gambar biner setelah melakukan proses erosi.

Dilation akan menaikkan ukurannya sehingga dapat menebalkan objek image dan menyambung object yang terputus ataupun meratakan tepi objek yang rusak. Proses selanjutnya setelah mendeteksi bentuk objek dengan menggunakan metode *find contour*.

[Figure 5 about here.]

Untuk syntax sebagai berikut:

```
Mat dilateElement = cv::getStructuringElement(MORPH_ELLIPSE, Size( 2 * nilaidilationbola + 1, 2 * nilaidilationbola + 1 ), Point( nilaidilationbola, nilaidilationbola ));
```

```
Ptr <cuda::Filter> dilateFilter = cuda::createMorphologyFilter(MORPH_DILATE, d.thresh.type(), dilateElement);
dilateFilter->apply(d.thresh, d.thresh);
```

E. Find Contour Method

Metode *findcontour* berguna untuk analisis bentuk dan deteksi dan pengenalan objek. Untuk mendeteksi bola maka *find contour* diatur agar hanya memproses bentuk lingkaran yang nantinya mendapat informasi area dan titik tengah suatu obyek. Dari hasil area dapat menggambar tanda lingkaran dengan area yang sesuai pada obyek yang terdeteksi sebagai pertanda bahwa *system* mendeteksi bola, dapat dilihat pada Figure 6. Sehingga *system* mengetahui posisi bola.

[Figure 6 about here.]

Metode *find contour* digunakan fungsi perintah `findContours` dari `OpenCV`. Perintah lengkap adalah sebagai berikut:

```
findContours(temp, contours, hierarchy, RETR_CCOMP, CHAIN_APPROX_SIMPLE);
approxPolyDP(contours[index], approxCircle, 0.05 * arcLength(Mat(contours[index]), true), true);
```

```
if (approxCircle.size() >= 5)
```

```
{
    objectFound = true
}
```

F. Jarak Bola

Untuk mencari bola penulis menggunakan rumus pythagoras. Dalil pythagoras atau rumus pythagoras berfungsi

untuk mencari salah satu sisi dengan kedua sisi diketahui. Rumus Pythagoras (Pythagoras) dapat dilihat pada Figure 7:

$$b^2 = a^2 + c^2$$

Maka untuk menghitung sisi tegak dan sisi mendatarnya berlaku rumus:

$$a^2 = b^2 -$$

$$c^2 \quad c^2 =$$

$$b^2 - a^2$$

Rumus Pythagoras dalam bentuk akar:

- Jika sisi miringnya c
- Sisi tegak dan mendatarnya adalah a dan b

[Figure 7 about here.]

Dibutuhkan library `#include <math.h>` Untuk dapat menjalankan perintah `sqrt` (akar) dan `pow` (pangkat).

```
const int FRAME_WIDTH = 640;
const int FRAME_HEIGHT = 480;
const Point CENTER_FRAME(FRAME_WIDTH / 2,
FRAME_HEIGHT / 2);
x = moment.m10 / moment.m00; // Koordiant titik x bola
y = moment.m01 / moment.m00; // Koordiant titik y bola
int sisib = CENTER_FRAME.x - x // sisi datar (b)
if (sisib < 0)sisib = sisib * -1; // agar nilai sisi datar (b) positif
int sisia = CENTER_FRAME.y - y; // sisi tegak lurus (a)
if (sisia < 0)sisia = sisia * -1; // agar nilai sisi tegak lurus (a) positif
sisic = sqrt(pow(sisia, 2) + pow(sisib, 2)); // // sisi miring (c)
```

[Figure 8 about here.]

Namun pada Figure 8 hasil dari sisi miring segita siku – siku yang diperoleh dari rumus Pythagoras masih berupa pixel, maka perlu mendata jarak bola terhadap kamera terbaca berapa pixel kemudian dari pengumpulan data tersebut dianalisis datanya menggunakan tool yang ada pada *software* Microsoft excel, analisis data yang digunakan adalah tipe regression, dari analisis tersebut akan mendapat nilai *coefficients*. Selanjutnya dua nilai *coefficients* yaitu *intercept* dan *x variable* 1 tersebut akan dijadikan parameter pada rumus untuk menkonversikan jarak pixel ke jarak cm. Persamaan dari rumus ini sebagai berikut:

$$\text{Input (jarak pixel)} = ((\text{jarak pixel}) * (x \text{ variable } 1) + \text{intercept}).$$

G. Sudut Arah Bola

Setelah mendapatkan nilai sisi miring dari sisi segitiga siku-siku yang diperoleh dari rumus pythagoras untuk parameter jarak berupa pixel, selanjutnya dikembangkan untuk mencari

sudut arah posisi bola. Diketahui pada Figure 9 persamaan untuk mencari sudut pada segita siku-siku sebagai berikut:

[Figure 9 about here.]

Pada gambar dibawah sudut yang dicari berada pada sudut A.

Persamaan untuk sudut arah bola adalah:

Derajat / Sudut A (α) = (sisi datar (AC) * sudut C) / sisi miring (AB)

$$\text{Derajat / Sudut A } (\alpha) = (\text{sisi b} * 90) / \text{sisi c}$$

Pada sistem yang dibuat nilai derajat mengikuti penggaris busur yaitu dari kiri memiliki nilai terendah 0 derajat hingga paling kanan memiliki nilai maksimal 180 derajat. Jadi jika posisi bola berada di kiri dari posisi *center* gambar, maka nilainya antara 0 - 90 derajat namun jika posisi bola berada di kanan dari posisi *center* gambar maka nilainya 90 – 180 derajat. Berikut ini syntax untuk membuat derajat 0 – 180:

```
const int FRAME_WIDTH =
640; const int FRAME_HEIGHT
= 480;
const Point CENTER_FRAME(FRAME_WIDTH / 2,
FRAME_HEIGHT / 2);
x = moment.m10 / moment.m00; // Koordiant titik x bola
y = moment.m01 / moment.m00; // Koordiant titik y bola
if (x < CENTER_FRAME.x) // Jika posisi bola dikiri
{
derajat = ((sisib * 90) / sisic);
}
Else //Jika posisi bola berada dikanan maka ditambah 90
{derajat = ((sisib * 90) / sisic) + 90}
```

HASIL DAN PEMBAHASAN

A. HSV

Tujuan dari proses ini adalah untuk menentukan nilai HSV yang memiliki parameter hue, saturation, dan value. Dari setiap parameter memiliki ambang batas minimal dan maksimal, jadi pada system ini memiliki 6 parameter yaitu:

- Minimal Hue
- Maksimal Hue
- Minimal Saturation
- Maksimal Saturation
- Minimal Value
- Maksimal Value

[Table 1 about here.]

Dari hasil Table I diketahui nilai HSV dengan hasil gambar biner bentuk bola yang baik untuk sistem terlihat pada no. 5 dengan nilai HSV yaitu, Hmin 18, Smin 189, Vmin 55, Hmax

35, Smax 255, Vmax 255.

B. Morfologis (Erosion and Dilation)

Tujuan dari proses ini adalah untuk menentukan nilai operasi *erosion* dan *dilation*. Sebab setelah dilakukan HVS *filter* ter- dapat noise, uji coba ini untuk menentukan nilai optimal dari *erosion* dan *dilation* sehingga dengan demikian, tingkat noise dapat ditekan tanpa harus mengorbankan luas wilayah warna atau ukuran dari objek *foreground*. Di bawah ini terdapat hasil dari 5 percobaan untuk mencari HSV yang tepat.

[Table 2 about here.]

Dari hasil Table II diketahui bahwa nilai *erosion* dan *dilation* yang baik untuk *system* terdapat pada no. 8 dengan nilai *erosion* 1 dan *dilation* 6.

C. Deteksi Berdasarkan Warna Objek

Untuk proses pengujian sistem deteksi warna bola yang digunakan adalah 4 buah bola dengan warna berbeda yaitu, hijau, putih, biru, dan *orange*. Bola diletakan dengan jarak yang sama, jarak yang akan diterapkan pada uji coba ini yaitu, 3 meter. Kamera yang digunakan adalah kamera Logitech. Tujuan pengujian ini untuk menguji HSV *filter* apakah berfungsi dengan baik dan dapat mendeteksi bola berwarna *orange*.

[Table 3 about here.]

Dari hasil Table III diketahui bahwa HVS *filter* berfungsi dengan baik dan *system* dapat mendeteksi bola berwarna *orange* walaupun terdapat beberapa bola warna berbeda pada sekitar bola *orange*.

D. Deteksi Berdasarkan Bentuk Objek

Proses ini menggunakan 2 objek dengan warna yang sama namun dengan bentuk berbeda, penulis menggunakan bentuk persegi dan lingkaran. 2 objek tersebut diletakan dengan jarak yang sama, jarak yang akan diterapkan pada uji coba ini yaitu, 1 meter hingga 8 meter. Kamera yang digunakan adalah kamera Logitech. Tujuan pengujian ini untuk melihat metode *find contour* apakah berfungsi dengan baik dan dapat mendeteksi bola berwarna *orange*.

[Table 4 about here.]

Dari hasil Table IV diketahui bahwa metode *find contour* berfungsi dengan baik dan *system* dapat mendeteksi bola berwarna *orange* pada jarak dibawah 6 meter, lebih dari 6 meter sistem terkadang mendeteksi objek lain bahkan tidak dapat membedakan bentuk benda, hal ini disebabkan karena objek

terlihat kecil jika terlalu jauh.

E. Deteksi Objek Terhalang

proses ini menggunakan objek bola berwarna *orange* dengan terhalangi oleh objek lain, penulis menghalangi objek bola warna *orange* dengan objek lain dengan setengah bagian dari bola *orange*. 2 objek tersebut diletakan dengan jarak yang sama, jarak yang akan diterapkan pada uji coba ini yaitu, 1 meter hingga 8. Kamera yang digunakan adalah kamera Logitech. Tujuan pengujian ini untuk melihat metode *find contour* apakah berfungsi dengan baik dan dapat mendeteksi bola berwarna *orange* yang terhalangi.

[Table 5 about here.]

Dari hasil Table V diketahui terdapat *system* mampu mendeteksi adanya bola (tetap mendeteksi bentuk lingkaran) dengan baik pada saat bola terhalang setengah oleh objek lain dengan jarak 1 hingga 9 meter. Dengan pencahayaan lampu yang cukup terang. Deteksi Jarak Bola Untuk mengetahui keakurasian sistem ini dalam menentukan jarak bola, maka dilakukan pengujian jarak bola pada sistem ini. Adapun hasil uji cobanya dapat dilihat dalam tabel ini.

[Table 6 about here.]

Dari hasil Table VI diketahui bahwa perbandingan antara jarak sebenarnya dengan jarak yang terbaca oleh sistem masih memiliki error, sistem ini memiliki tingkat akurasi dalam membaca jarak sebesar 92.87 persen.

F. Deteksi Sudut Bola

Pengujian ini dilakukan di laboratorium elektronika Hang Tuah Surabaya dengan posisi kamera horizontal terhadap objek. Bola yang digunakan adalah bola futsal berwarna *orange*. Sudut yang akan diuji yaitu 0 – 180 derajat.

[Table 7 about here.]

Dari hasil Table VII diketahui bahwa implementasi rumus Pythagoras untuk mencari sudut bola berfungsi namun masih terdapat *error* dan memiliki tingkat akurasi 88.19%, untuk mendapatkan nilai *error* digunakan rumus sudut sebenarnya dikurangi sudut terbaca, kemudian dari hasil tersebut dibagi sudut sebenarnya.

$$\text{Error} = \frac{\text{Sudut} - \text{Sudut terbaca}}{\text{Sudut}} \times 100\%$$

G. DETEKSI BOLA DENGAN KECEPATAN BERVARIASI

Untuk mendapatkan data tentang kecepatan bola, maka di-

lakukan pengujian kecepatan bola. Pengujian dilakukan dengan panjang lintasan bola 6 meter dengan menghitung waktu tempuhnya. Tujuan dari percobaan ini adalah untuk menentukan kecepatan bola yang dapat dideteksi oleh kamera. Berikut ini adalah tabel tentang kecepatan bola yang dapat terdeteksi oleh kamera.

Accepted: 2020-03-31
Published: 2020-06-13

[Table 8 about here.]

Dari data Table VIII diatas, maka dapat disimpulkan bahwa kecepatan bola apabila makin lambat maka akan mudah dideteksi oleh kamera dan terlihat semakin jelas. Kemudian bila bola semakin cepat maka akan sulit terdeteksi oleh kamera karena tangkapan gambar menjadi semakin kurang jelas. Namun dari hasil tabel diatas bola tetap terdeteksi pada kecepatan 10m/s walaupun sistem mendeteksi bola terlambat (*delay*).

KESIMPULAN

Dari hasil pengujian dan pengambilan data yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Dengan pencahayaan ruangan yang cukup terang Dengan pencahayaan ruangan yang cukup terang
2. Sistem tetap dapat mendeteksi bola yang terhalang setengah bagian oleh objek lain pada jarak hingga 9 meter.
3. Sistem mampu memberikan informasi jarak bola dalam satuan centimeter dengan tingkat akurasi 92.89 persen.
4. Sistem mampu memberikan informasi sudut posisi arah bola dengan tingkat akurasi 88.19 persen.
5. Sistem mampu mendeteksi objek bola.

REFERENSI

- [1] J. Subur, T. A. Sardjono, & R. Mardiyanto, "Braille Character Recognition Using Find Contour and Artificial Neural Network," *International Journal of Electrical and Electronics Engineering*, vol. 14, no. 1, pp. 19–24, 2016.

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Author [s]. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Received: 2020-03-13

DAFTAR TABEL

I	Data nilai HSV color filtering dan hasil	143
II	Data gambarbiner dengan nilai erosion 0 – 1, dan nilai dilation 0 – 6	143
III	Data deteksi bola orange dengan 4 bola berbedawarna berjarak 3 meter	143
IV	Data deteksibola orange pada 2 objek benda berwarna orange dengan bentuk berbeda berjarak 1 – 8 meter.....	144
V	Data deteksibola orange yang terhalang oleh objek lain denganjarak 1 – 9 meter	144
VI	Data sudut derajat bola.	144
VII	Data sudut derajat bola.	145
VIII	Tabel pengujiansistem deteksi bola bergerak dengan kecepatan 2 m/s hingga 10 m/s	145

TABLE I. DATA NILAI HSV COLOR FILTERING DAN HASIL

No	H low	S low	V low	H up	S up	V up	Hasil
1	14	201	90	22	255	217	
2	11	207	82	24	255	255	
3	12	30	79	30	255	255	
4	15	26	100	30	255	255	
5	18	189	55	35	255	255	

TABLE II. DATA GAMBARBINER DENGAN NILAI EROSION 0 – 1, DAN NILAI DILATION 0 – 6

No.	Nilai Erosion	Nilai Dilation	Gambar	Keterangan
1	0	0		Terdapat noise di luar dan di dalam objek (objek terdapat bintik hitam)
2	1	0		Noise di luar objek berkurang, namus noise di dalam objek tetap (objek terkikis)
3	1	1		Masih terdapat noise di dalam objek (objek terkikis)
4	1	2		Masih terdapat noise di dalam objek (objek terkikis)
5	1	3		Masih terdapat noise di dalam objek (objek terkikis)
6	1	4		Masih terdapat noise di dalam objek (objek terkikis)
7	1	5		Masih terdapat noise di dalam objek (objek terkikis)
8	1	6		Tidak ada noise

TABLE III. DATA DETEKSI BOLA ORANGE DENGAN 4 BOLA BERBEDA WARNA BERJarak 3 METER.

No.	Percobaan	Warna	Gambar	Keterangan
(1)	(2)	(3)	(4)	(5)
1	2 bola	Biru Orange		Terdeteksi (Berhasil)
2	2 bola	Putih		Terdeteksi (Berhasil)
		Orange		
3	2 bola	Hijau		Terdeteksi (Berhasil)
		Orange		
4	2 bola	Hijau		Tidak
		Biru		Terdeteksi (Berhasil)
5	3 bola	Hijau		Tidak
		Biru		Terdeteksi (Berhasil)
		Putih		
6	4 bola	Orange		Terdeteksi (Berhasil)
		Hijau		
		Biru		
		Putih		

TABLE IV. DATA DETEKSIBOLA ORANGE PADA 2 OBJEK BENDA BERWARNA ORANGE DENGAN BENTUK BERBEDA BERJARAK 1 – 8 METER

No.	Percobaan	Jarak (Meter)	Gambar	Keterangan
(1)	(2)	(3)	(4)	(5)
1	2 objek berbeda bentuk berwarna orange	1 meter		Mendeteksi bola
2	2 objek berbeda bentuk	2 meter		Mendeteksi bola
3	2 objek berbeda bentuk	3 meter		Mendeteksi bola
4	2 objek berbeda bentuk	4 meter		Mendeteksi bola
5	2 objek berbeda bentuk	5 meter		Mendeteksi bola
6	2 objek berbeda bentuk	6 meter		Mendeteksi bola
7	2 objek berbeda bentuk	7 meter		Mendeteksi bola namun terkadang mendeteksi objek berbentukkotak
8	2 objek berbeda bentuk	8 meter		Mendeteksi objek berbentuk kotak

TABLE V. DATA DETEKSIBOLA ORANGE YANG TERHALANG OLEH OBJEK LAIN DENGANJARAK 1 – 9 METER.

No.	Percobaan	Jarak (Meter)	Gambar	Keterangan
(1)	(2)	(3)	(4)	(5)
1	Bola terhalang setengah	1 meter		Terdeteksi
2	Bola terhalang setengah	2 meter		Terdeteksi
3	Bola terhalang setengah	3 meter		Terdeteksi
4	Bola terhalang setengah	4 meter		Terdeteksi
5	Bola terhalang setengah	5 meter		Terdeteksi
6	Bola terhalang setengah	6 meter		Terdeteksi
7	Bola terhalang setengah	7 meter		Terdeteksi
8	Bola terhalang setengah	8 meter		Terdeteksi
9	Bola terhalang setengah	9 meter		Terdeteksi

TABLE VI. DATA SUDUT DERAJAT BOLA.

No.	Jarak (cm)	Jarak Terbaca (cm)	Erorr (%)	Akurasi (%)
(1)	(2)	(3)	(4)	(5)
1	100	98	2.00%	98
2	200	192	4.00%	96
3	300	315	5.00%	95
4	400	430	7.50%	92.5
5	500	540	8.00%	92
6	600	645	7.50%	92.5
7	700	780	11.43%	88.57
8	800	870	8.75%	91.25
9	900	990	10.00%	90
Rata - rata			7.13%	92.87

TABLE VII. DATA SUDUT DERAJAT BOLA.

No. (1)	Jarak (m) (2)	Sudut (3)	Sudut terbaca (4)	Erorr (%) (5)	Akurasi (%) (6)
1	1	65	55	18.18%	81.82%
2	2	84	70	20.00%	80.00%
3	3	90	90	0.00%	100.00%
4	4	124	110	12.73%	87.27%
5	5	143	125	14.40%	85.60%
6	1	90	70	28.57%	71.43%
7	2	105	90	16.67%	83.33%
8	3	125	110	13.64%	86.36%
9	4	144	125	15.20%	84.80%
10	5	59	55	7.27%	92.73%
11	1	90	90	0.00%	100.00%
12	2	125	110	13.64%	86.36%
(1)	(2)	(3)	(4)	(5)	(6)
13	3	145	125	16.00%	84.00%
14	4	60	55	9.09%	90.91%
15	5	79	70	12.86%	87.14%
16	1	115	110	4.55%	95.45%
17	2	145	125	16.00%	84.00%
18	3	60	55	9.09%	90.91%
19	4	80	70	14.29%	85.71%
20	5	90	90	0.00%	100.00%
21	1	136	125	8.80%	91.20%
22	2	65	55	18.18%	81.82%
23	3	80	70	14.29%	85.71%
24	4	90	90	0.00%	100.00%
25	5	123	110	11.82%	88.18%
Rata – rata				11.81%	

TABLE VIII. TABEL PENGUJIANSISTEM DETEKSI BOLA BERGERAK DENGAN KECEPATAN 2 M/S HINGGA 10 M/S

No. (1)	Kecepatan (m/s) (2)	Gambar (3)	Keterangan (4)
1	2.13		Terdeteksi bola terlihat jelas
2	3.85		Terdeteksi bola terlihat jelas
3	6.67		Terdeteksi bola sedikit blur
4	8.33		Terdeteksi bola sedikit semakin blur
5	10.00		Terdeteksi namun terlambat bola sedikit sangat blur

DAFTAR GAMBAR

1	Diagram blok perancangan sistem	147
2	Konversi bingkai dari RGB ke HSV	147
3	Trackbar, ouput threshold dan click drag rectangle.	147
4	Gambar kiri HSV filter, gambar kanan output proses erosi.	147
5	Gambar kiri erosi, gambar kanan output proses dilasi.	148
6	Tracking Object, pada benda berbentuk lingkaran.	148
7	Rumus Pythagoras dalam bentuk akar	148
8	Jarak Bola dengan rumus Pythagoras	149
9	Persamaan untuk mencari sudut σ pada segitiga siku – siku	149

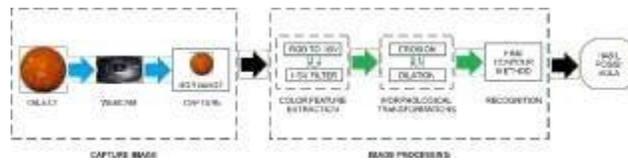


Figure 1. Diagram blok perancangan sistem

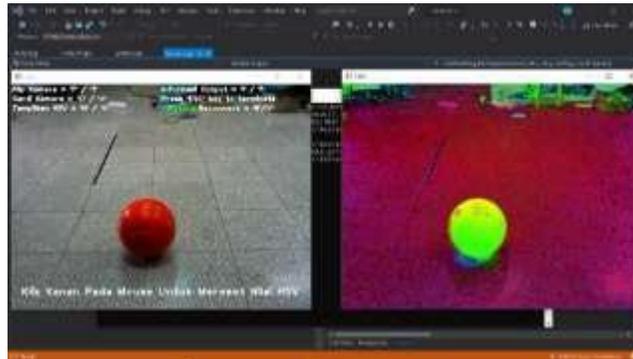


Figure 2. Konversi bingkai dari RGB ke HSV.

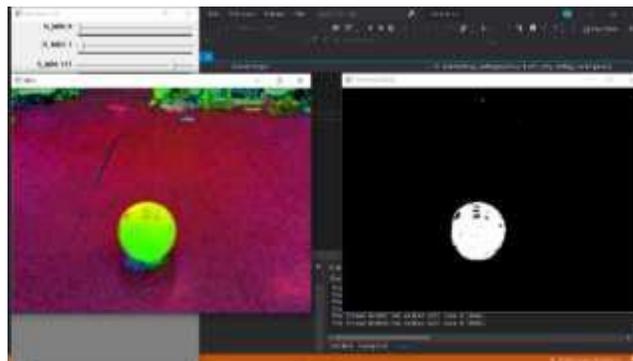


Figure 3. Trackbar, output threshold dan click drag rectangle.

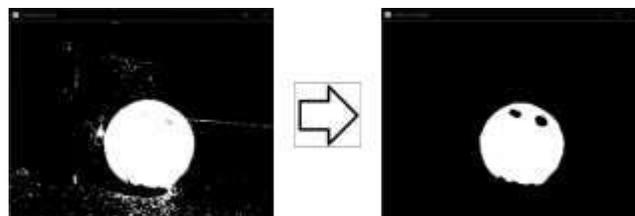


Figure 4. Gambar kiri HSV filter, gambar kanan output proses erosi.

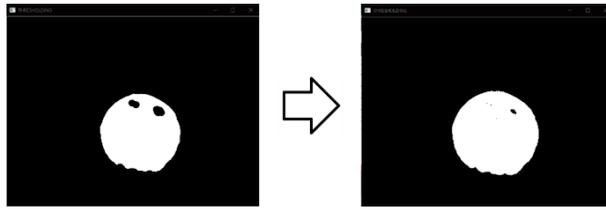


Figure 5. Gambarkiri erosi, gambar kanan output proses dilasi.

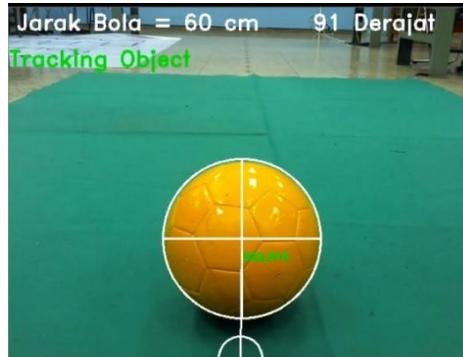


Figure 6. Tracking Object, pada benda berbentuk lingkaran.

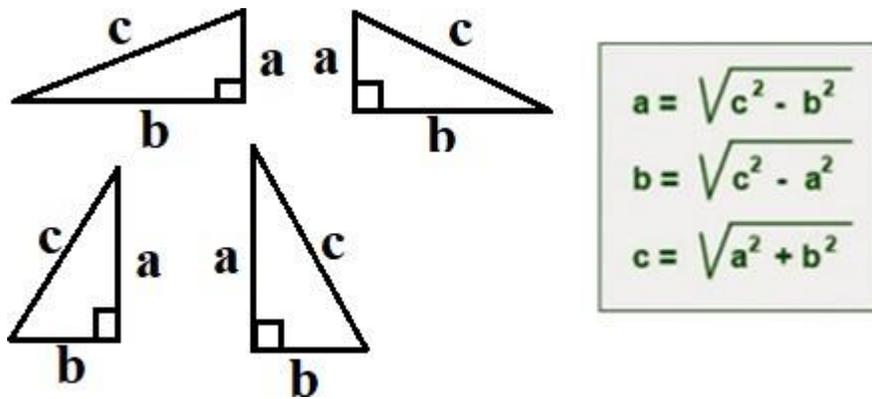


Figure 7. Rumus Pythagoras dalam bentuk akar.

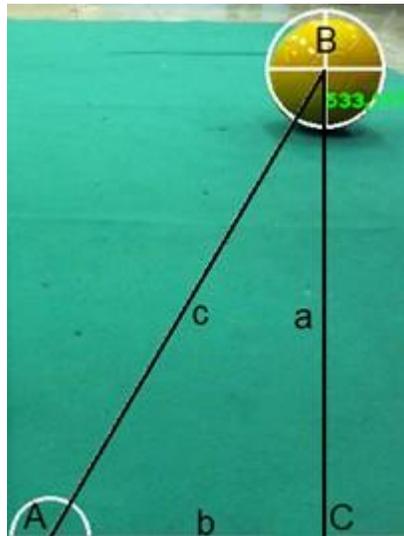
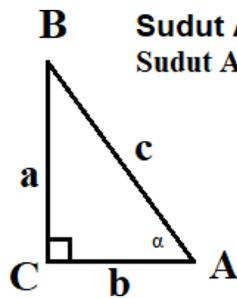
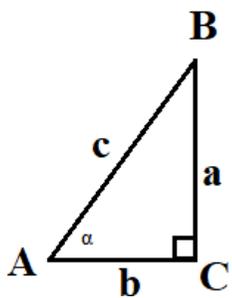


Figure 8. Jarak Bola dengan rumus Pythagoras.



Sudut A (α) = (sisi datar (b) * sudut siku-siku (C)) / sisi miring (c)
Sudut A (α) = (sisi b * 90) / sisi c

Figure 9. Persamaan untuk mencari sudut α pada segitiga siku – siku.