

## DETEKSI OBJEK BERWARNA MENGGUNAKAN METODE COLOR MATCHING BERBASIS ARDUINO

Dhany Satya Utama<sup>1)</sup>, Basuki Rahmat<sup>2)</sup>, Wahyu Syaifullah Jauharis Saputra<sup>3)</sup>

E-mail: <sup>1)</sup>satyadhany@gmail.com, <sup>2)</sup>basukirahmat.if@upnjatim.ac.id,

<sup>3)</sup>wahyu.s.j.saputra@gmail.com

<sup>1 2 3</sup>Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional  
"Veteran" Jawa Timur

### Abstrak

Akhir – akhir ini banyak ajang perlombaan dalam dunia robotika dimana kamera menjadi salah satu peranan penting dalam perlombaannya. Hal itu memerlukan pengolahan citra pada hasil tangkapan kamera agar robot dapat berjalan mengikuti objek yang berada didepannya. Beberapa penelitian menggunakan banyak metode untuk melakukan pengolahan citra. Ada yang menggunakan metode bentuk objek untuk memfokuskan pada bentuknya, namun metode ini kurang maksimal karena banyak objek dalam lapangan yang bentuknya sama. Maka dari itu menggunakan metode selanjutnya yaitu metode *Color Matching* untuk melakukan pencocokan warna pada sebuah objek agar kamera dapat lebih terfokus pada objek dengan warna tertentu. Pengolahan citra dengan metode *Color Matching* akan ditanamkan pada Arduino untuk memudahkan pengguna melakukan inisialisasi warna awal. Penangkapan gambar menggunakan kamera serial OV7670 yang cocok untuk memulai penelitian, karena pembacaan *pixel* yang tidak terlalu cepat. Uji coba yang dilakukan pada sistem pendeteksi objek berwarna ini salah satunya adalah pengujian warna objek dan latar belakang. Warna yang didapat harus berada dalam intensitas warna yang ditentukan, ini dimaksudkan untuk pendeteksian yang maksimal. Intensitas warna yang semakin besar menyebabkan *noise* yang semakin banyak pula, yang juga menyebabkan pergerakan alat ini menjadi tidak teratur.

**Kata kunci:** Kamera Serial OV7670, Color Matching, Arduino.

## 1 PENDAHULUAN

Akhir-akhir ini banyak ajang perlombaan dalam dunia robotika dimana kamera menjadi salah satu peran penting dalam perlombaannya. Dalam perlombaan itu adalah ajang dimana robot dapat menemukan object tertentu secara autonomus, dan hal itu memerlukan pengolahan citra agar kamera dapat memfokuskan pada benda yang telah ditentukan dan akan memberi informasi agar robot dapat jalan sesuai arahan dari pengolahan citra tersebut. Untuk itu adanya sebuah aplikasi simulasi robot pendeteksi yang dilengkapi dengan kamera agar dapat memudahkan pengembangan dari robot pendeteksi citra tersebut. Penggunaan kamera yang sesuai adalah hal penting untuk menangkap gambar yang optimal. Kamera OV7670 adalah sensor *CMOS* yang memproduksi berbagai format melalui *Serial Camera Control Bus (SCCB)* [1].

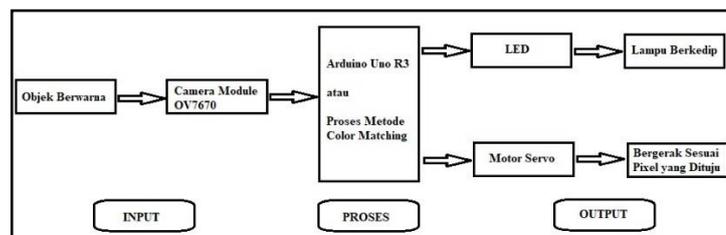
Simulasi robot pendeteksi ini dibuat menggunakan mikrokontroler sebagai pusat pengendali dari robot tersebut [2], dimana mikrokontroler yang digunakan adalah Arduino UNO. Arduino adalah sebuah mikrokontroler yang mudah untuk dikembangkan, karena bahasanya yang sederhana seperti Bahasa C [3]. Arduino ini memiliki cara kerja menggunakan digit biner, sehingga cocok untuk melakukan pendeteksian gambar dengan *pixel – pixel* untuk memudahkan untuk melakukan pemograman. Banyak cara yang dilakukan untuk melacak suatu objek, cara yang paling populer adalah melakukan deteksi menggunakan warna RGB sebagai ukuran pendeteksian. Dikarenakan warna RGB adalah warna – warna dasar dari segala objek dalam kehidupan sehari – hari dan juga warna yang

sering digunakan dalam pengolahan citra digital [4]. Sehingga diperlukan metode untuk pencocokan warna atau *Color Matching* untuk mendeteksi warna RGB tersebut agar dapat mengikuti atau mendeteksi objek bergerak didepannya [5]. Penggerak menggunakan motor servo, karena Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat di-*setup* atau diatur untuk menentukan dan memastikan posisi sudut dari poros *output* motor [6].

## 2 METODOLOGI

### 2.1 Blok Diagram

Awal perancangan ini akan dimulai dengan membahas blok diagram dari sistem ini agar mudah untuk membuat rancangan selanjutnya. Secara umum blok diagram dari alat pendeteksi objek berwarna adalah sebagai berikut:

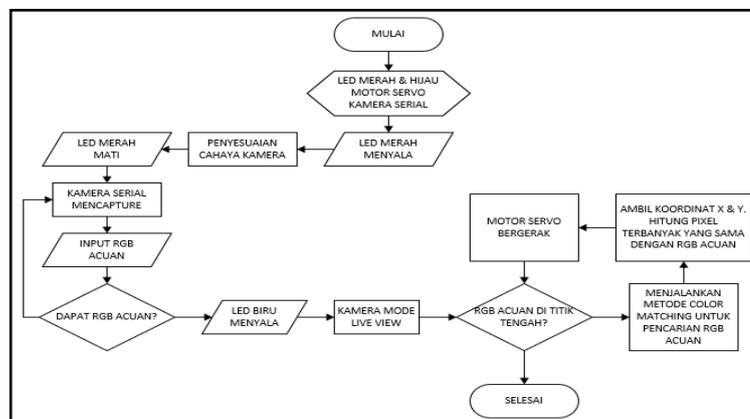


Gambar 1. Blok Diagram Deteksi Objek

Langkah pertama mengambil gambar yang akan di ambil 1 *pixel* tengah sebagai acuan warna. Lalu kamera akan berjalan layaknya *live view* (langsung) dan akan diambil gambar per *millisecond* untuk dilakukan pencocokan warna menggunakan Metode *Color Matching* dengan warna acuan tersebut. Jika warna acuan tersebut tidak pada titik tengah maka kamera akan bergerak menggunakan motor servo hingga ke arah warna acuan yang terbanyak. Lampu LED yang menjadi indikator yang menunjukkan bahwa kamera serial sedang melakukan pengambilan gambar serta sedang melakukan proses *Color Matching*.

### 2.2 Flowchart Sistem

Pada bagian ini akan dibahas bagaimana algoritma sistem yang akan dibuat dengan menggunakan *Flowchart*. Berikut adalah gambaran algoritma dari sistem pendeteksi objek berwarna ini.



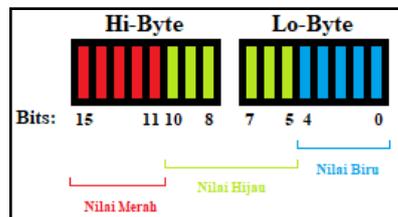
Gambar 2. Flowchart Sistem Deteksi Objek

Pada simbol “Mulai” menandakan alat dalam keadaan menyala. Kemudian simbol “Inialisasi” menandakan proses inialisasi komponen yang terdapat pada sistem pendeteksi objek berwarna yaitu Kamera Serial OV7670, Lampu LED, dan dua Motor

Servo. Ketika proses inialisasi selesai LED berwarna merah akan menyala untuk menandakan kamera belum siap digunakan, hal ini terjadi karena kamera harus menyesuaikan pencahayaan pada lingkungan terlebih dahulu. Setelah itu LED berwarna merah akan mati dan mulai mengambil gambar untuk memasukkan *input*-an warna RGB acuan. Jika warna RGB acuan tidak didapatkan, maka kamera kembali mengambil gambar hingga mendapatkan *input*-an warna RGB acuan. Jika sudah dapat, maka LED berwarna biru akan menyala yang menandakan proses Pendeteksi Objek Berwarna sudah dimulai. Kamera Serial OV7670 dalam keadaan *live view* yang akan mengambil gambar secara terus menerus, dan setiap *frame* atau gambar di akan disamakan dengan warna RGB acuan. Jika warna RGB acuan tidak berada pada koordinat X dan Y yang seharusnya, maka akan menjalankan proses Metode *Color Matching* untuk mencari koordinat X dan Y dari warna *pixel – pixel* yang sesuai dengan warna RGB acuan. Jika koordinat X dan Y sudah didapatkan maka akan dikirim datanya ke motor servo agar bergerak menuju ke arah koordinat X dan Y yang ditentukan.

### 2.3 Proses Color Matching

Untuk proses pengambilan warna, Kamera Serial OV7670 menghasilkan *output* berupa RGB565, dimana warna yang dihasilkan berupa 2 byte, *Highbyte* dan *Lowbyte*.



Gambar 3. RGB 565

Sehingga warna RGB565 tersebut akan di convert menjadi RGB888 pada umumnya agar lebih mudah waktu proses *color matching*. Pertama gabung dahulu *Highbyte* dan *Lowbyte* menjadi 16 digit binary dengan rumus:

$$colorByte = (colorHigh \ll 8) \text{ OR } colorLow \quad (1)$$

Rumus tersebut menjelaskan bahwa digit *Highbyte* akan digeser ke kiri sebanyak 8 digit yang menjadikan 8 digit pertama diisi oleh *Highbyte*. Kemudian pakai rumus gerbang logika OR untuk menghubungkan digit *Lowbyte*. Table kebenaran gerbang OR adalah sebagai berikut.

$$R = \left( \left( \left( (colorByte \gg 11) \text{ AND } 31 \right) \times 527 \right) + 23 \right) \gg 6 \quad (2)$$

Perhitungan warna *Red* yaitu hasil gabungan RGB565 (*colorByte*) akan digeser sebanyak 11 digit binary. Lalu akan di benarkan dengan gerbang logika AND dengan angka 31 (untuk bilangan desimal), dan dikalikan 527 lalu ditambah 23 dan digeser lagi sebanyak 6 digit.

$$R = \left( \left( \left( (colorByte \gg 11) \text{ AND } 31 \right) \times 527 \right) + 23 \right) \gg 6 \quad (3)$$

Perhitungan warna *Green* yaitu hasil gabungan RGB565 (*colorByte*) akan digeser sebanyak 5 digit binary. Lalu akan di benarkan dengan gerbang logika AND dengan angka 63 (untuk bilangan desimal), dan dikalikan 259 lalu ditambah 33 dan digeser lagi sebanyak 6 digit.

$$B = \left( \left( (colorByte \text{ AND } 31) \times 527 \right) + 23 \right) \gg 6 \quad (4)$$

Perhitungan warna *Blue* yaitu hasil gabungan RGB565 (*colorByte*) akan langsung di benarkan dengan gerbang logika AND dengan angka 31 (untuk bilangan desimal)

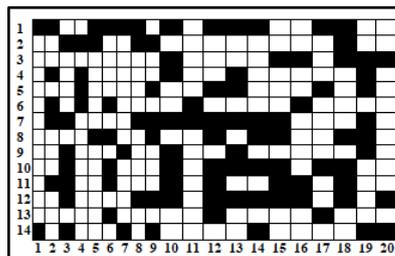
dikarenakan ini adalah digit terakhir dalam RGB565. Dan dikalikan 527 lalu ditambah 23 dan digeser lagi sebanyak 6 digit.

Setelah RGB565 selesai diubah menjadi RGB888, maka akan dicari pencocokan warnanya. Untuk kamera OV7670 sudah peka terhadap perubahan cahaya yang masuk, jadi kendalanya adalah warna yang sama tetapi karena terkena pencahayaan yang berbeda warna tersebut sedikit memudar ataupun sedikit lebih gelap. Pada jurnal [4] menjelaskan tentang hasil penelitian pada rentang intensitas beberapa warna dasar, dapat dilihat pada tabel berikut ini.

**Tabel 1. Rentang instensitas warna RGB pada warna primer dan sekunder.**

Warna	Rentang Warna		
	Red	Green	Blue
Hijau	0 - 173	100 – 255	0 - 170
Biru	0 – 240	0 – 248	112 – 255
Merah	128 – 255	0 – 160	0 – 128
Kuning	102 – 255	102 – 255	0 - 50
Magenta	75 – 255	0 – 230	128 – 255
Cyan	0 – 224	128 – 255	20 – 255

Intensitas warna tersebut akan dijadikan refrensi sementara yang nantinya akan diuji coba secara langsung menggunakan alat pendeteksi yang akan dibuat. Dari rentang intensitas warna yang ditemukan dapat disimpulkan rata – rata *range* atau jaraknya antara 167, jika dibagi menjadi 2, jarak ke lebih terang sekitar 83 dan jarak ke lebih gelap sekitar 83 juga. Setelah mengetahui cara menghitung kecerahan dan kegelapan warna tersebut, maka selanjutnya akan mencari jumlah *pixel* terbanyak dari salah satu baris *pixel* yang nantinya akan menjadi *pixel* utama dalam proses *Color Matching* ini. Perhitungannya adalah dengan membandingkan berapa jumlah *pixel* yang sama antar baris. Contohnya jika ada *pixel* dengan warna seperti gambar berikut ini.

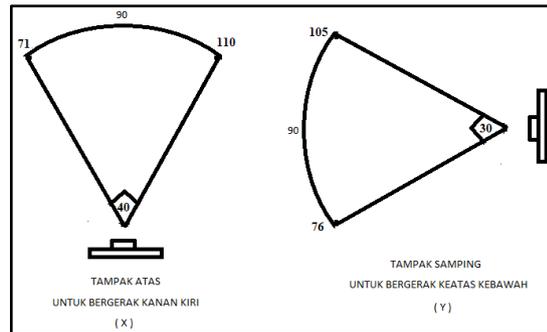


**Gambar 4. ColorMatching pada *pixel* replica**

Seperti gambar di atas, setelah program menemukan kesamaan warna maka akan ditandai dengan tanda warna Hitam, jika tidak maka berwarna Putih. Pertama melakukan perbandingan antar kolom, dimana kolom 1 dan kolom selanjutnya harus sama – sama memiliki kesamaan warna dengan warna acuan, jika tidak maka akan dihitung jumlah kolom yang sama, dan akan dibandingkan dengan jarak kolom – kolom berikutnya yang memiliki kesamaan dengan warna acuan. Dicari yang paling besar *range* atau jarak kolom yang sama, yang nantinya akan menjadi kolom *point* sementara.

Lalu menuju ke baris berikutnya. Baris berikutnya pun sama mencari *range* atau jarak kolom yang sama dengan warna acuan, lalu dibandingkan dengan baris sebelumnya. Begitupun seterusnya hingga menemukan baris dari *range* kolom terpanjang, dan menjadikannya baris *point* serta kolom *point*. Jika kolom *point*-nya panjang, maka akan diambil salah satu *pixel* ditengahnya yang akan dijadikan kolom *point*. Ketika sudah menemukan baris *point* dan kolom *point* yang sama dengan warna acuan, maka selanjutnya

akan dihitung untuk menentukan pergerakan motor servo. Dalam *pixel* Kamera Serial OV7670, baris *point* sama seperti koordinat Y dan kolom *point* sama seperti koordinat X.

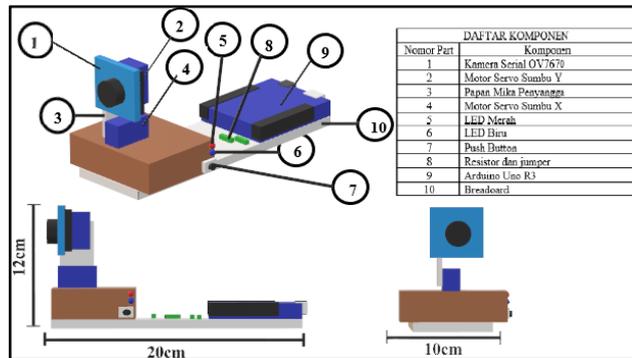


Gambar 5. Pergerakan motor servo

Untuk pergerakan motor servo sendiri menggunakan sudut tengah  $90^\circ$  untuk titik tengahnya, dan akan bergerak  $1^\circ$  apabila perbedaannya 4 *pixel*. Jika pada Kamera Serial sumbu X berjumlah 160 *pixel* dan sumbu Y berjumlah 120 *pixel*. Maka akan dibagi 4 agar pergerakan motor servo sesuai dengan *pixel* kamera, menjadi  $40^\circ$  untuk sumbu X dan  $30^\circ$  untuk sumbu Y. Perbandingannya seperti 40:160 untuk servo X : sumbu X, dan 30:120 untuk servo Y : sumbu Y.

#### 2.4 Rancangan Alat

Sebelum melakukan pembuatan perlu diketahui rancangan dari alat yang akan dibuat agar memudahkan saat merangkai alat – alat tersebut. Rancangan alat dapat dilihat pada gambar berikut ini.

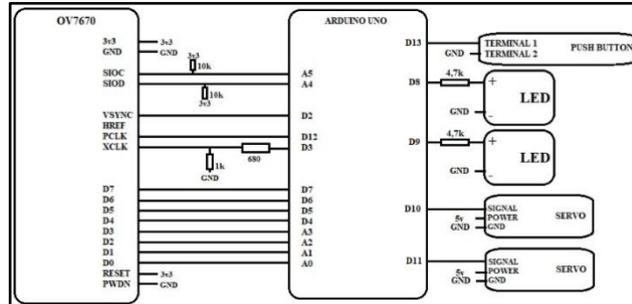


Gambar 6. Rancangan Alat

Alat ini terdiri dari beberapa komponen utama yang diperlukan. Kamera serial OV7670 akan ditempatkan pada sisi teratas agar dapat melihat objek yang berada di depannya. Kamera serial OV7670 tersebut disatukan dengan motor servo sumbu Y untuk menggerakkan ke atas dan ke bawah. Lalu dibuatkan penyangga agar kamera tidak berbenturan dengan motor servo sumbu X yang berfungsi sebagai rotasi ke samping. Pada *plater* terdapat 2 buah LED berwarna merah dan biru sebagai penanda, serta *push button* untuk memudahkan pengguna mengetahui penanda dan menekan tombolnya. Lalu pada papan bawah terdapat *breadboard* untuk meletakkan komponen resistor dan *jumper – jumper*. Bagian belakang terdapat Arduino Uno R3 agar mudah memasukkan program dengan kabel USB.

## 2.5 Rangkaian Alat

Pada subbab ini akan menjelaskan tentang bagaimana cara pemasangan alat elektronika berbasis mikrokontroler Arduino Uno. Skematik rangkaian dan konfigurasi pin dapat dilihat pada gambar berikut ini.

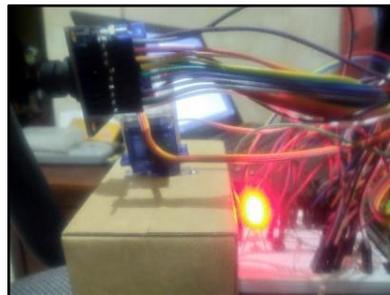


Gambar 7. Skematik rangkaian

Komponen yang digunakan pada pembuatan alat ini adalah komponen yang sudah disusun sedemikian rupa untuk menunjang pembuatan alat ini. Komponen ini berbentuk *module* dan yang biasa. Jika semua komponen sudah terpasang sesuai dengan konfigurasi pin – pin nya, maka tahap selanjutnya yaitu pemrograman mikrokontroler.

## 3 HASIL DAN PEMBAHASAN

Alat pendeteksi objek berwarna ini pada dasarnya mulai bekerja sesaat setelah pengguna menekan tombol untuk mengaktifkan mode pendeteksian objek berwarna, lalu input akan diteruskan ke Arduino Uno kepada komponen – komponen lainnya.



Gambar 8. Proses pendeteksian

Warna pertama akan masuk ke *HighByte* kemudian ke *LowByte*, begitupun seterusnya. *HighByte* dan *LowByte* akan disatukan menjadi 16 bit seperti rumus yang dijelaskan pada rumus (1). Lalu warna tersebut dirubah ke dalam satuan 8bit dari masing – masing warna RGB. Jika mode deteksi belum menyala maka sistem akan terus mengambil titik warna titik tengah untuk dijadikan *RGB point*. Jika mode deteksi dimulai maka sistem akan membandingkan warna dari *frame* yang didapat dengan warna *RGB point* dengan nilai batas atau *threshold* 83 untuk mendapatkan warna yang lebih gelap ataupun warna cerahnya.

Perhitungan untuk motor servo menggunakan modulus 2 untuk jarak antar *pixel* utama dan *pixel* baru agar motor servo dapat menyesuaikan pergerakannya dan tidak berlebihan dalam bergerak. Setelah itu mencari koordinat sumbu Y dengan membandingkan baris utama dengan baris yang didapat untuk mengetahui kemana arah dari pergerakan motor servo, bergerak ke atas atau bergerak ke bawah. Jika baris yang didapatkan lebih kecil maka motor servo bergerak ke atas dan membatasinya dengan sudut

60°, jika baris yang didapatkan lebih besar maka motor servo bergerak ke bawah dan membatasinya dengan sudut 120°.

Pada pencarian posisi sumbu X pun sama membandingkan kolom utama dengan kolom yang didapatkan. Jika kolom yang didapatkan lebih kecil daripada kolom utama maka motor servo akan bergerak ke kiri dan membatasinya dengan sudut 50°, jika lebih besar maka motor servo akan ke kanan dan membatasinya dengan sudut 130°. Jika hasil yang didapat melewati batas – batas tersebut maka sumbu X ataupun sumbu Y akan dijadikan batas *pixel* terpinggir dari *frame* tersebut. Setelah selesai, motor servo akan berjalan sesuai titik hasil koordinat yang didapatkan. Alat pendeteksi ini akan terus mengulang hingga adaptor atau tegangan yang mengalir ke Arduino Uno dilepaskan.

### 3.1 Pengujian Kamera Serial OV7670

Pengujian ini bertujuan untuk mengetahui berapa resolusi gambar yang bisa ditangkap oleh kamera serial OV7670 serta berapa waktu yang dibutuhkan untuk menangkap 1 *frame* gambar. Hasil pengujian dapat dilihat pada tabel berikut ini.

Tabel 2. Pengujian Kamera Serial OV7670

Percobaan ke -	Resolusi Gambar	Baud	Waktu yang dibutuhkan	Hasil Gambar
1	160x120	115200	6 detik	
2	160x120	1000000	2 detik	
3	320x240	1000000	4 detik	

Hasil uji coba berdasarkan Tabel 2 Pengujian Kamera Serial OV7670 menjelaskan tentang *frame* yang dapat diambil oleh kamera serial OV7670 yang menggunakan mikrokontroler Arduino Uno dan waktu yang dibutuhkan juga berbeda – beda antara 2 – 6 detik/*frame*. Hal ini dikarenakan kemampuan *processing* dari IC Arduino Uno yaitu Atmega 328 terbatas. Oleh karena itu jika menggunakan Arduino Uno dengan kamera serial OV7670 gunakan resolusi gambar pada ukuran 160x120(VGA) dengan *baud* 115200 yang membutuhkan waktu 6 detik.

Pada percobaan kedua dan ketiga benar gambar lebih besar dan waktu pun lebih cepat dibandingkan dengan yang pertama, tetapi gambar yang dihasilkan pun menjadi rusak atau tidak jelas. Hal ini dikarenakan *image processing* pada Arduino Uno tidak mampu mengikuti kecepatan 1.000.000 *baud* yang ada pada rule kamera serial OV7670 dan *pixel*-pun menjadi terpotong – potong. Jadi direkomendasikan jika menggunakan mikrokontroler Arduino Uno dan kamera serial OV7670 sebaiknya menggunakan resolusi 160x120 dengan 115200 *baud* karena sudah cukup jelas untuk menangkap objek berwarna.

### 3.2 Pengujian Jarak Pendeteksi Warna

Pengujian ini bertujuan untuk mengetahui jarak minimal dan maksimal dari alat pendeteksi ini dapat mendeteksi warna tersebut. Hasil pengujian dapat dilihat pada tabel berikut ini.

**Tabel 3. Pengujian jarak kamera untuk pendeteksian warna**

No	Jarak (cm)	Gambar Objek	Terdeteksi?
1	<50		Terdeteksi
2	50 – 100		Terdeteksi dengan sedikit <i>noise</i>
3	>100		Sulit, banyak <i>noise</i>

Berdasarkan hasil pengujian sistem yang berada pada ruangan dengan pencahayaan yang cukup seperti Tabel 3 menghasilkan *output* pendeteksian yang baik dijarak dibawah 50cm, pencahayaan nya cukup dan pengambilan warnanya jelas. Pada jarak 50cm hingga 100cm menghasilkan *noise* sedikit yang mengakibatkan kamera OV7670 tidak dapat mendeteksi dengan hasil yang maksimal. Jarak lebih dari 100cm akan sulit mendeteksi objek yang berwarna karena warna cahaya semakin berbeda sehingga menghasilkan *noise* yang cukup besar.

### 3.3 Pengujian Pencahayaan Pendeteksi Warna

Pengujian ini bertujuan untuk mengetahui seberapa bisanya alat pendeteksi ini mendeteksi objek berwarna meskipun sudah diberikan *threshold* atau ambang batas warna. Hasil pengujian dapat dilihat pada tabel berikut ini.

**Tabel 4. Pengujian pecahaya**

Nilai Cahaya	Kondisi Lingkungan	Gambar Objek	Terdeteksi?	Keterangan
0 lux – 1 lux	Gelap		Tidak	Tidak ada cahaya mempengaruhi kamera sehingga hasil pendeteksian menjadi sangat buruk
2 lux – 30 lux	Samar--samar		Iya, dengan sedikit <i>noise</i>	Minimnya cahaya mempengaruhi kamera mendapatkan hasil warna RGB yang sempurna
31 lux – 15000 lux	Terang		Iya	Nilai cahaya yang ideal untuk penelitian menggunakan kamera serial OV7670 sehingga pendeteksian berjalan normal
>15000 lux	Sangat terang		Iya, dengan sedikit <i>noise</i>	Cahaya berlebihan membuat pengaruh besar bagi warna

Hasil dari pengujian pencahayaan pada Tabel 4 dapat dijelaskan bahwa pencahayaan yang cukup dapat menghasilkan *output* yang maksimal, pada nilai cahaya ketiga yaitu 31 lux – 15000 lux merupakan cahaya ideal untuk melakukan penelitian menggunakan kamera serial OV7670. Dibandingkan dengan nilai cahaya ideal tersebut nilai 0 – 1 lux menjadikan kamera tidak dapat mengenali warna karena terlalu gelap. Nilai 2 – 30 lux sebenarnya dapat mendeteksi objek berwarna, namun terlalu banyak *noise* dikarenakan minimnya cahaya yang masuk. Nilai diatas 15000 lux juga dapat melakukan pendeteksian, tetapi cahaya berlebih membuat warna berubah – ubah, apalagi jikalau objek berwarna tersebut bergerak maka warna yang dihasilkan akan tidak sesuai dengan *threshold* yang dimasukkan.

### 3.4 Pengujian Warna Pencahayaan Lampu

Pengujian ini menggunakan pencahayaan penerangan lampu dalam suatu ruangan. Lampu yang digunakan adalah lampu led berwarna putih dan bohlam berwarna kuning dengan nilai cahaya antara 31 lux – 15000 lux. Berikut adalah tabel hasil pengujiannya.

**Tabel 5. Pengujian Warna Pencahayaan Lampu**

Jenis Lampu	Hasil
Lampu lilin putih	Sukses Terdeteksi
Bola lampu kuning	Terdeteksi dengan sedikit <i>noise</i>

Hasil dari Tabel 5 menunjukkan bahwa lampu penerangan berwarna putih sangat cocok untuk penelitian kamera serial OV7670 dikarenakan warna yang dihasilkan sama dengan warna dari objek tersebut. Penerangan lampu warna kuning juga cocok, tetapi warna yang dihasilkan objek akan lebih sedikit mengarah ke warna kuning karena efek dari lampu penerangan tersebut.

### 3.5 Pengujian Warna Objek dan Latar Belakang

Pengujian ini menggunakan latar belakang warna cerah dan gelap. Warna cerah menggunakan warna putih dan warna gelap menggunakan warna coklat tua. Objek berwarna akan diuji pada ruangan yang memiliki nilai cahaya antara 31 lux – 15000 lux. Hasil pengujian dapat dilihat pada tabel berikut ini.

**Tabel 6. Hasil pengujian warna objek dan latar belakang**

Warna	Gambar Objek	Terlacak?	Keterangan
Objek Merah dan LB Putih		Ya	Sukses terdeteksi
Objek Hijau dan LB Putih		Ya	Sukses terdeteksi
Objek Biru dan LB Putih		Ya	Sukses terdeteksi
Objek Abu – Abu dan LB Putih		Tidak	Gagal terdeteksi karena <i>threshold</i> yang begitu mirip
Objek Merah Muda dan LB Putih		Ya	Terdeteksi dengan sedikit <i>noise</i>
Objek Ungu dan LB Putih		Ya	Sukses terdeteksi
Objek Kuning dan LB Putih		Ya	Terdeteksi dengan sedikit <i>noise</i>
Objek Merah dan LB Coklat		Ya	Terdeteksi dengan sedikit <i>noise</i>
Objek Abu – Abu dan LB Coklat		Ya	Sukses terdeteksi
Objek Merah Muda dan LB Coklat		Ya	Sukses terdeteksi
Objek Ungu dan LB Coklat		Ya	Terdeteksi dengan sedikit <i>noise</i>

Hasil dari Tabel 6 tersebut dapat dijelaskan bahwa alat pendeteksi ini mengenali warna objek apabila *threshold* dan warna latar belakang terpaut jauh, sehingga kamera

serial OV7670 dapat membedakan warna objek dengan latar belakang yang ada. Hasil tersebut dapat mendeteksi berbagai warna, tetapi penulis menggunakan warna yang ada pada lingkungan sekitar untuk melakukan pengujian.

## 4 KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian tugas akhir yang telah dilakukan adalah mikrokontroler Arduino Uno dapat mengimplementasikan sistem pendeteksian objek berwarna menggunakan metode *Color Matching*. Dimana Atmega328 yang mempunyai ukuran memori yang cukup kecil ternyata dapat menampung data – data *pixel* hasil dari pengambilan gambar kamera serial OV7670 dan sekaligus dapat menggerakkan motor servo secara bersamaan.

Pengolahan citranya sendiri mengacu pada warna RGB yang berada pada koordinat sumbu X dan Y yang telah ditentukan. Pencarian warna yang tepat dilakukan dengan cara mencari perbandingan antar kolom, dimana kolom 1 dan kolom selanjutnya harus mempunyai warna yang sama dengan warna RGB acuan, dan akan dibandingkan jumlah jarak kolom terbanyak. Selanjutnya pada bagian baris dari citra pun juga akan dilakukan perbandingan jumlah jarak kolom yang didapat yang akan dihitung untuk mencari koordinat sumbu X dan Y yang baru.

Pengujian terakhir robot pendeteksi warna objek ini dilakukan dengan mengubah warna objek dan latar belakang yang berbeda – beda. Hasilnya alat pendeteksi warna ini mampu untuk mendeteksi dari berbagai macam warna, meskipun waktu yang dibutuhkan bisa dibilang tidak cepat. Hal ini dikarenakan pengolahan citra pada Atmega328 cukup lambat yang memproses *pixel* satu per satu sehingga saat objek tersebut berpindah tempat hasil tangkapan warna *pixel* yang dihasilkan akan berubah ditengah – tengah pengambilan gambar.

### 4.2 Saran

Sistem pendeteksi objek berwarna menggunakan metode *Color Matching* berbasis mikrokontroler Arduino ini masih sangat banyak kekurangan, baik itu dari segi *hardware* maupun *software*-nya, oleh sebab itu penulis menyarankan kepada penelitian selanjutnya, yaitu:

- a. Pengembangan konsep dari alat ini dapat ditambahkan sebuah sistem pemantauan gambar langsung agar pengguna tidak kesulitan saat menggunakannya.
- b. Melakukan penelitian ini dengan menggunakan sensor kamera yang berbeda dan menggunakan mikrokontroler Arduino yang memiliki kinerja yang cepat agar mendapatkan hasil yang maksimal.
- c. Tambahkan fitur – fitur yang lebih menarik, dari fitur *hardware* maupun *software*-nya, untuk menarik para peminat dalam bidang robotika serta untuk melakukan riset yang ada pada komunitas Robotika yang berada pada naungan Program Studi Informatika UPN “Veteran” Jatim.

## 5 DAFTAR RUJUKAN

- [1] Chitturi, O., & Rajasekaran, A., 2017. FPGA Based Camera Control System . *International Journal for Scientific Research & Development*, 5(5), pp.1626-1629.
- [2] Güven, Y., Coşgun, E., Kocaoğlu, S., Gezici, H., & Yilmazlar, E., 2017. Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications. *Research Inveny: International Journal of Engineering And Science*, 6(9), pp.38-44.

- [3] Louis, L., 2016. Working Principle Of Arduino And Using It As A Tool For Study And Research. *International Journal of Control, Automation, Communication and Systems (IJACS)*, 1(2), pp.21-29.
- [4] Yudha, Y., Ardhiyanta, D., Haris, L., & Widiarti, A. R., 2016. Aplikasi Pengenalan Citra Warna Dasar. *Jurnal Ilmiah Widya Teknik*, 15(1), pp. 54-57.
- [5] Ajie , S., 2018. Analisis Proses Color Matching Warna Spesial Hijau Toska Dengan Penambahan Tinta Medium Pada Kemasan X Sesuai Customer Approval. *Jurnal Risenologi Kpm Unj*, 3(1), pp.31-36.
- [6] Rusmida., 2015. Rancang Bangun Nampan Keseimbangan. *Jurnal Ilmiah Mikrotek*, 1(4), pp.106-113.