

PERANCANGAN APLIKASI PEMBACA WARNA DAN BENTUK BERBASIS PENGOLAHAN CITRA UNTUK DAFTAR KATALOG PERPUSTAKAAN

Design Of Color And Shape Reader Application On Image Processing Application For Library Catalog List

Annisa' Sulistyowati¹, Yuli Sun Hariyani ,ST.,MT.², Atik Novianti,S.ST.,MT.³

^{1,2}Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Telkom University

¹annisaslst@gmail.com, ²yulisun@tass.telkomuniversity.ac.id, ³atiknovianti@tass.telkomuniversity.ac.id

Abstrak

Berkembangnya sebuah teknologi pengolahan citra digital (*Digital Image Processing*) yang semakin pesat, yang dapat mempermudah kehidupan manusia, terdapat aplikasi yang dapat menerapkan pengolahan citra itu sendiri, dalam berbagai bidang. Jika kita ingin mengembalikan sebuah buku di perpustakaan biasanya menggunakan teknologi *barcode reader*. Untuk itu, dibuatlah sebuah aplikasi yang serupa dengan *barcode*, aplikasi ini untuk membaca warna dan bentuk untuk sistem informasi perpustakaan berbasis pengolahan citra untuk membantu petugas dalam hal estimasi waktu dalam mengelola buku di perpustakaan.

Pada Proyek Akhir ini telah dibuat sebuah aplikasi yang dapat membaca warna dan bentuk yang kemudian disimpan di sebuah database yang telah dibuat, dalam pengerjaan ini menggunakan *library opencv*. Cara kerja dari aplikasi ini adalah, *webcam* dipasang untuk *capture* gambar pada punggung buku, lalu secara otomatis sistem akan mendeteksi dari warna dan bentuk buku tersebut

Dari hasil pengujian dari aplikasi pembaca warna ini didapatkan keberhasilan mendeteksi warna sebesar 100% dan untuk keberhasilan dari mendeteksi bentuk sebesar 66% berdasarkan cahaya ideal pada ruangan 70 lux sampai 150 lux dan jarak 5 cm. Jarak ideal untuk mendeteksi sebesar 5cm sampai 10 cm, rata-rata waktu proses yang diperoleh sistem untuk mendeteksi warna dan bentuk adalah 1.67 detik.

Kata kunci : Pengolahan Citra, *Opencv*, *Perpustakaan*

Abstract

The development of the digital image processing technology (Digital Image Processing) that is increasingly fast, which can facilitate human life, there are applications that can apply image processing itself, in various fields. If we want to return a book to the library, we usually use barcode reader technology. For that, an application is made that is similar to a barcode, this application is for reading and forms for information systems that can be used to help workers in matters that require time in managing books in the library.

In this Final Project, an application has been created that can read colors and shapes which are then stored in a database that has been created using the opencv library. The working of this application is, the webcam is installed to capture images on the back of the book, the results of the detection will be entered in the book code to be matched with the database.

From the results of testing of this color reader application, it is found that the success of detecting colors by 100% and for the success of detecting the shape is 66% based on ideal light in rooms 70 lux until 150 lux and a distance of 5 cm. The ideal distance for detecting is 5cm to 10 cm, the average processing time is obtained by the system to detect colors and shapes is 1.67 seconds.

Keywords : *Image Processing, OpenCV, Library*

1. Pendahuluan

Seiring perkembangan teknologi Pengolahan Citra yang semakin berkembang, di Indonesia sudah mengalami kemajuan yang pesat. Dengan memanfaatkan kemajuan ilmu teknologi pengolahan citra digital (*Digital Image Processing*), maka dapat mempermudah proses sistem di Perpustakaan. Beberapa perpustakaan memakai teknologi *barcode reader* untuk sistem Peminjaman dan pengembalian buku. Aplikasi yang telah dibuat serupa dengan teknologi *Barcode*. Pada Proyek Akhir ini akan mengimplementasikan pengolahan citra untuk sebuah sistem di perpustakaan menggunakan *Library OpenCV*, untuk medianya menggunakan *webcam*

Beberapa penelitian terkait, Irfan Chandra S[2], yang mengimplementasikan deteksi warna dengan *Orange Pi* dan *library OpenCV*. Pada Proyek akhir ini akan dibuat sebuah program yang bisa mendeteksi warna merah, kuning, hijau, biru, dan untuk bentuk lingkaran, segitiga, persegi. Pada penelitian Rio Pradhitya [7] dalam judul Pembangunan Aplikasi Deteksi dan Tracking Warna *Virtual Drawing* Menggunakan Algoritma *Color Filtering*. Pada penelitian tersebut pendeteksian warna menggunakan *color filtering* yang penerapannya masih bersifat statis hanya warna menggunakan merah saja, dan penelitian tersebut masih berupa simulasi program. Oleh karena itu pada Proyek Akhir ini akan dibuat sebuah program yang dapat diterapkan secara nyata.

Pada penelitian Eka Adhiyanto, Wiwien Hardikurniawati dan Zuli Budiarmo[3] dalam judul Implementasi Metode *Image Subtracting* dan Metode *Regionprops* untuk Mendeteksi Jumlah Objek Berwarna RGB pada file Video, warna yang dapat dikenali adalah warna objek yang merupakan warna pokok, sehingga perlu untuk dikembangkan lagi. Dirancangnya sistem pembaca warna untuk perpustakaan, petugas perpustakaan dapat terbantu dengan adanya aplikasi ini. Prinsip kerjanya adalah, petugas perpustakaan hanya memegang buku yang sudah ada tanda (bentuk dan gambar) kemudian arahkan ke kamera *webcam*, kemudian sistem akan langsung membaca bentuk dan warna (misal : Bentuk *Circle* warna Merah, maka sistem akan menampilkan jenis buku tersebut).

2. Dasar Teori

2.1 Pengolahan Citra

Pengolahan Citra (*Image Processing*) adalah sebuah pemrosesan citra menggunakan computer, menjadi citra yang kualitasnya lebih baik dan sesuai dengan keinginan pemakai. Tujuan dari *Image Processing* adalah memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau komputer[2]. Pengolahan Citra mengubah citra ke citra yang lainnya, masukannya berupa citra dan keluaran berupa citra juga yang akan lebih baik daripada citra masukannya.

Pengolahan citra memiliki beberapa fungsi, diantaranya adalah :

1. Digunakan untuk memperbaiki kualitas citra agar lebih mudah diinterpretasi oleh manusia ataupun computer.
2. Digunakan untuk Teknik untuk Teknik pengolahan citra dengan mentransformasikan citra menjadi citra lain. Contoh : pemampatan citra (*image compression*)
3. Sebagai proses awal (*preprocessing*) dari komputer visi (*computer vision*)[4]

2.2 Visual Studio

Microsoft Visual Studio by merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Compiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

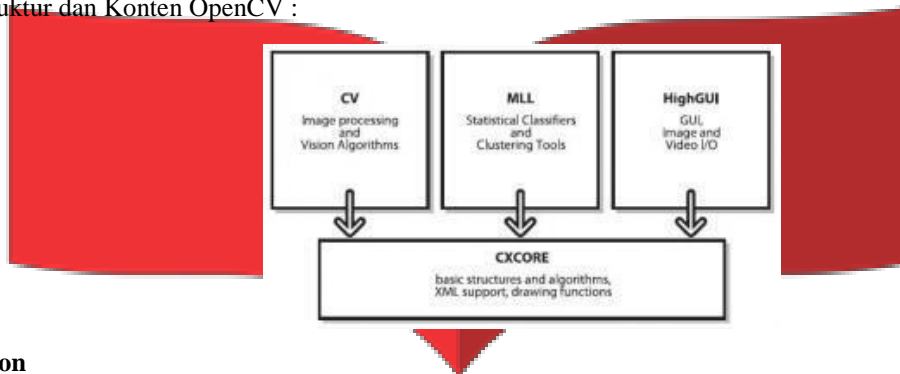
2.3 OpenCv

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision. Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, dll. OpenCV adalah library *Open Source* untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video.

OpenCV sendiri terdiri dari 5 library, yaitu :

3. CV : untuk algoritma Image processing dan Vision.
4. ML : untuk machine learning library
5. Highgui : untuk GUI, Image dan Video I/O.
6. CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.
7. CvAux

Struktur dan Konten OpenCV :



2.4 Python

Python adalah salah satu bahasa pemrograman yang sering terdengar ditelinga kita. Untuk menggunakan bahasa pemrograman ini sebelumnya harus diartikan kedalam sesuatu sebelum program dijalankan. Bahasa pemrograman ini memiliki beberapa kelebihan yaitu lebih singkat dalam menulis sebuah program sehingga dapat menghemat waktu. Kelebihan selanjutnya yaitu program lebih pendek dan mudah dibaca. Dan kelebihan yang terakhir yaitu *portable* yang berarti Python dapat dijalankan pada komputer yang berbeda dengan sedikit atau tanpa modifikasi[7].

2.5 Ruang Warna

Ruang warna (juga disebut model warna atau sistem warna) bertujuan untuk memudahkan menspesifikasikan warna dalam beberapa standar. Model warna adalah kordinat dan subsferensi di dalam sistem dimana masing – masing warna diwakili oleh satu titik.

2.5.1 Ruang warna RGB

Ruang warna RGB merupakan ruang warna dari 3 warna dasar, yaitu merah(*red*), hijau(*green*), dan biru(*blue*). Ketiga warna ini dapat dikombinasikan menjadi beragam warna baru. Masing-masing warna ini memiliki jumlah intensitas yang sama, yaitu 8-bit (0-255). Citra RGB terdiri dari 3 buah layer yang merepresentasikan array matriks warna dengan \square baris \times \square kolom yang kemudian disatukan. Maka dari itu setiap piksel dalam array tersebut terdiri dari 3 komponen warna yaitu merah (R), hijau (G), dan biru (B) dengan nilai yang berbeda-beda[1]

2.5.2 Ruang Warna HSV (Hue, Saturation, Value)

Ruang warna HSV (Hue Saturation Value) menunjukkan ruang warna dalam tiga komponen yaitu *Hue*, *Saturation* dan *Value*. *Hue* adalah sudut 0 sampai 360 derajat yang menunjukkan posisi warna pada spektrum warna. *Saturation* mempunyai nilai 0 sampai 100 yang menunjukkan tingkat keabuan warna. Dan *Value* mempunyai nilai 0 sampai 100 yang menunjukkan tingkat kecerahan warna[1].

2.6 SQLite

Sqlite Merupakan perpustakaan perangkat lunak yang menerapkan engine database SQL secara mandiri, tanpa memerlukan server, tanpa perlu melakukan konfigurasi, dan bersifat transaksional. SQLite adalah engine database SQL yang paling banyak digunakan di dunia[8]

2.7 Webcam

Kamera Web atau yang dikenal dengan istilah *Webcam* adalah sebutan bagi kamera waktu nyata yang gambarnya bisa dilihat melalui www (World Wide Web), program pengolah pesan cepat, atau aplikasi pemanggilan video. Istilah *webcam* merujuk pada teknologi secara umumnya, sehingga kata *webcam* kadang-kadang diganti dengan kata lain yang memberikan pemandangan yang ditampilkan di kamera, misalnya *StreetCam* yang memperlihatkan pemandangan jalan ada juga *Metrocam* yang memperlihatkan pemandangan panorama kota dan pedesaan, *TraffiCam* yang digunakan untuk memonitor keadaan jalan raya, cuaca dengan *WeatherCam*, bahkan keadaan gunung berapi dengan *VolcanoCam*. Kamera web dapat diartikan juga sebagai sebuah kamera video digital kecil yang dihubungkan ke komputer melalui port USB, port COM atau dengan jaringan Ethernet atau Wi-Fi[1].

3. Perancangan Sistem

Gambar 3.1 adalah blok diagram atau alur kerja dari aplikasi deteksi objek warna untuk pengambilan bentuk citra.

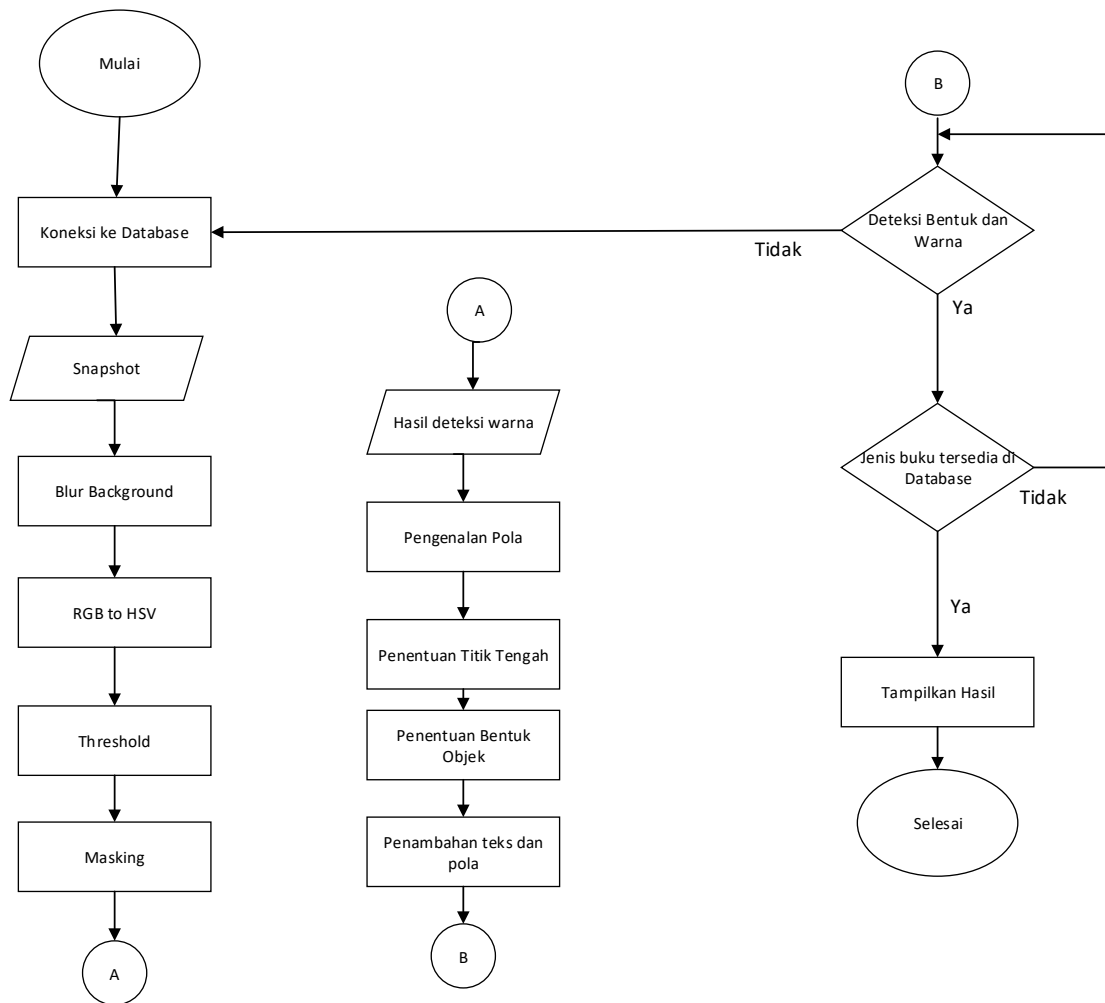


Gambar 3. 1 Blok Diagram

Berdasarkan blok diagram seperti terlihat pada gambar 3.1 terdapat beberapa tahapan dalam implementasi yakni sebuah *software* pendeteksi warna dan bentuk menggunakan, *Library OpenCV* sebagai pendukung agar dapat menjalankan program berbasis computer vision sistem ini dapat mendeteksi warna dan bentuk sebuah buku. Perangkat keras yang digunakan yaitu berupa kamera yang berfungsi untuk menangkap gambar yang akan dideteksi kemudian untuk memproses gambar yang telah didapat dan dapat di tampilkan hasil proses gambar pada monitor.

Pada bab ini telah dijelaskan mengenai perancangan sistem aplikasi pembaca warna yang terdiri dari sub bab analisa kebutuhan sistem, realisasi sistem, skenario pengujian dan parameter dalam pengujian. Adapun alur perancangan sistem pembaca warna yang telah disimulasikan dapat dilihat pada Gambar 3.2 di bawah ini.

3.1 Flowchart Sistem



Gambar 3. 2 Flowchart deteksi warna dan bentuk

Berdasarkan Gambar 3.2 flowchart perancangan sistem telah dilakukan dengan tahap-tahap sebagai berikut :

1. Koneksi Database

Semua buku disimpan didatabase untuk memudahkan pencocokan pada jenis warna dan bentuk pada buku tersebut.

2. Blur Background

Pada tahap ini dilakukan pengebluran background untuk memisahkan objek yang akan *dicapture* . Fungsi dari *Blur* yaitu untuk menghaburkan gambar. Pada tahap ini cukup membantu dalam menghilangkan noise kecil, sehingga mempermudah pendeteksian

3. RGB to HSV

Setelah proses *Blur* tahap selanjutnya adalah mengkonversi gambar inputan yang berformat RGB ke gambar yang berformat HSV, karena gambar yang berformat HSV lebih mudah untuk memfilter ruang warna.

4. Threshold

Pada tahap ini ditentukan lower dan upper. Fungsi lower dan upper ini untuk

mendapatkan nilai batas atas dan batas bawah dari sebuah objek gambar yang akan dideteksi untuk nantinya menjadi nilai threshold pada sistem.

5. Masking

Setelah didapatkan hasil dari gambar dari *Threshold*, lalu gambar dikembalikan lagi ke warnanya untuk pixel bernilai 1 untuk mengetahui objek tersebut hanya berwarna merah biru dan hijau.

6. Pengenalan Pola

Setelah hasil didapatkan pada deteksi warna, langkah selanjutnya adalah pengenalan pola, sebelumnya sistem tidak dapat langsung menentukan objek berada dimana, berwarna apa, dan berbentuk apa. Didalam library OpenCV terdapat perintah untuk dapat mengenali pola – pola yang ada seperti perintah berikut

7. Penentuan titik tengah

Setelah mendapatkan pola yang terdapat pada objek, langkah selanjutnya adalah mencari titik tengah pada objek. Untuk menentukan titik tengah dari objek kita harus mendapatkan moment dari proses pengenalan pola.

8. Penentuan Bentuk Objek

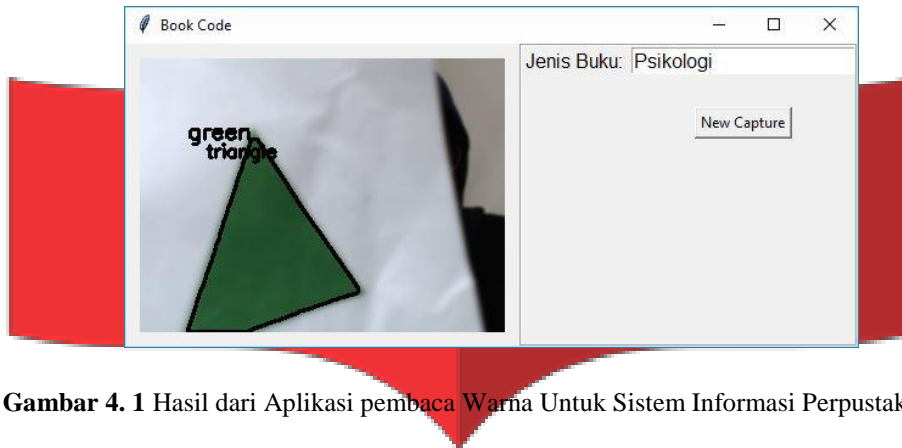
Setelah mendapatkan titik tengah dan sudut – sudut yang ada pada objek langkah selanjutnya memperkenalkan sistem kepada bentuk – bentuk yang akan dideteksi yaitu lingkaran, segitiga, dan persegi. Jika pada sebuah objek mempunyai sudut sebanyak 4 maka objek tersebut dikenali oleh sistem berbentuk persegi. Dan jika pada sebuah objek mempunyai sudut sebanyak 3 maka objek tersebut dikenali oleh sistem berbentuk segitiga.

9. Penambahan teks dan Pola

Langkah selanjutnya pada sistem ini adalah dengan menambahkan teks dan gambar pola yang ada pada objek. setelah sistem mengetahui warna dan bentuk pada objek tersebut sistem tidak dapat memberitahukan secara langsung warna dan bentuk pada objek tersebut maka dengan menambahkan teks dan bentuk garis yang mengelilingi objek akan membantu sistem dalam memberitahukan warna dan bentuk pada objek

4. Hasil Aplikasi

Aplikasi yang dibuat untuk mengetahui informasi dari sebuah buku di perpustakaan. Dari hasil gambar 4.1 dibawah ini mendeteksi sebuah objek dengan warna hijau yang berbentuk segitiga. Setelah warna dan bentuk terdeteksi langkah selanjutnya adalah menginputkan kode jenis buku tersebut. Kode buku dari *green triangle* adalah 236, setelah menginputkan kode buku lalu muncul sebuah informasi dari jenis buku tersebut.



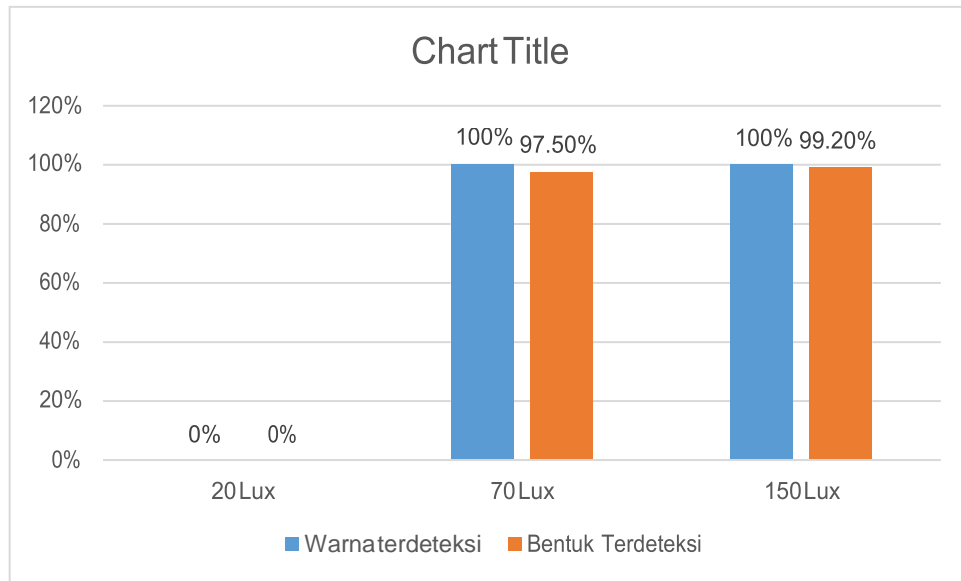
Gambar 4. 1 Hasil dari Aplikasi pembaca Warna Untuk Sistem Informasi Perpustakaan

4.1 Pengujian Pengaruh Cahaya

Pada pengujian ini dilakukan berdasarkan kondisi cahaya pada ruangan. Dimana parameter ini dibagi menjadi 3 kondisi, yang pertama kondisi ketika cahaya pada ruangan 20 lux, 70 lux dan 150 lux, pengujian ini dilakukan pada jarak 5 cm.

Tabel 4. 1 Hasil Pengujian Lux

Nilai Lux	Warna	Bentuk	Hasil Warna	Hasil Bentuk	Presentase		
					Warna	Bentuk	
20	Merah	△	Tidak Terdeteksi	-	-	-	
		□	Tidak Terdeteksi	-	-	-	
		○	Tidak Terdeteksi	-	-	-	
	Hijau	△	Tidak Terdeteksi	-	-	-	
		□	Tidak Terdeteksi	-	-	-	
		○	Tidak Terdeteksi	-	-	-	
		Biru	△	Tidak Terdeteksi	-	-	-
			□	Tidak Terdeteksi	-	-	-
			○	Tidak Terdeteksi	-	-	-
	70	Merah	△	Terdeteksi	△	100%	90%
			□	Terdeteksi	□	100%	98%
			○	Terdeteksi	○	100%	95%
Hijau		△	Terdeteksi	△	100%	100%	
		□	Terdeteksi	□	100%	100%	
		○	Terdeteksi	○	100%	95%	
Biru		△	Terdeteksi	△	100%	100%	
		□	Terdeteksi	□	100%	100%	
		○	Terdeteksi	○	100%	100%	
100		Merah	△	Terdeteksi	△	100%	98%
			□	Terdeteksi	□	100%	95%
			○	Terdeteksi	○	100%	100%
	Hijau	△	Terdeteksi	△	100%	100%	
		□	Terdeteksi	□	100%	100%	
		○	Terdeteksi	○	100%	100%	
	Biru	△	Terdeteksi	△	100%	100%	
		□	Terdeteksi	□	100%	100%	
		○	Terdeteksi	○	100%	100%	



Gambar 4. 2 Grafik Perbandingan Akurasi Berdasarkan Cahaya

Hasil pengujian deteksi objek berdasarkan nilai lux didapatkan hasil akhir dari 20 lux terdapat nilai keberhasilan deteksi warna sebesar 0%, dan bentuk 0% karena terlalu gelap. Untuk nilai lux sebesar 70 lux terdapat nilai keberhasilan dalam mendeteksi warna sebesar 100% dan untuk mendeteksi bentuk sebesar 97.5%. untuk nilai lux sebesar 150 lux didapatkan hasil bahwa untuk deteksi objek terdapat keberhasilan mendeteksi warna sebesar 100% dan untuk mendeteksi bentuk sebesar 99.2%.

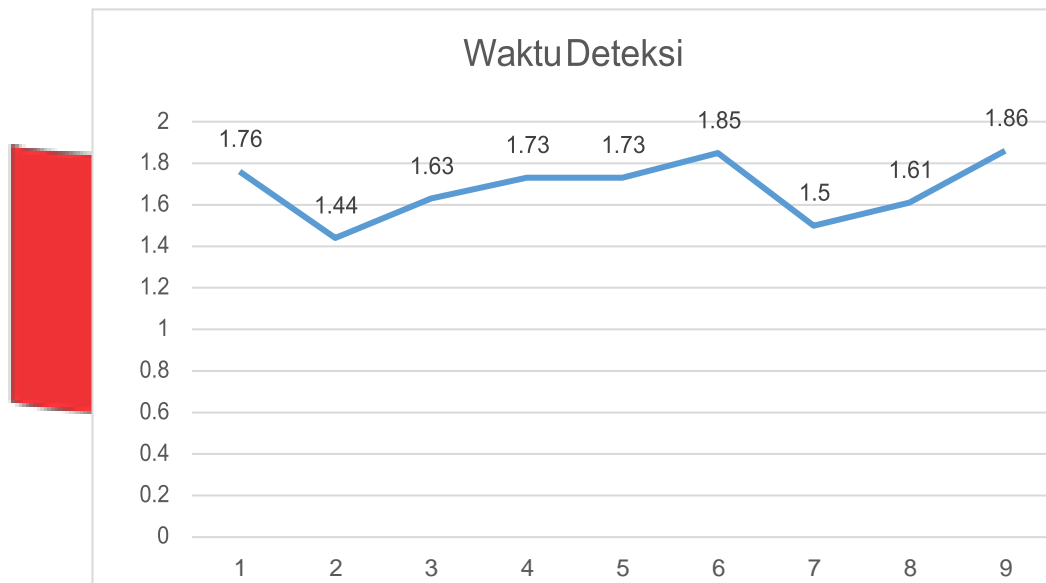
4.2 Pengujian Berdasarkan Waktu Proses Deteksi Objek

Pengujian berdasarkan waktu proses dari deteksi objek adalah membandingkan waktu sistem mendeteksi gambar yang berbeda. Pengujian ini dilakukan untuk mengetahui pengaruh objek yang menentukan dimanakah jarak pengambilan yang baik untuk sistem agar dapat mendeteksi warna dan bentuk objek. Pengujian ini dilakukan dengan beberapa parameter yaitu :

- Dengan cahaya sebesar 100 lux
- Objek yang terdeteksi adalah 3 bentuk dan 3 warna.
- Menggunakan nilai blur (5,5)

Tabel 4. 2 Hasil Pengujian Berdasarkan Waktu

No	Warna	Bentuk	Waktu
1	Merah	△	1.76
2		□	1.44
3		○	1.63
4	Hijau	△	1.73
5		□	1.73
6		○	1.85
7	Biru	△	1.5
8		□	1.61
9		○	1.86
Rata-rata waktu deteksi Objek			1.67



Gambar 4.3 Grafik waktu proses deteksi

$$\text{Nilai rata-rata Waktu Proses Deteksi Objek} = \frac{\sum \text{waktu proses}}{\text{Total Pengujian}} = \frac{15.09}{9} = 1.67 \text{ detik}$$

Hasil pengujian waktu deteksi objek dapat dilihat pada table 4.2 didapatkan hasil akhir nilai lamanya sistem mendeteksi objek yaitu 1.67 detik. Perubahan waktu dari tiap percobaan tidak terlalu signifikan dan cenderung stabil.

4.3 Pengujian Berdasarkan Jarak Pengujian Gambar

manakah jarak pengambilan gambar yang baik untuk sistem agar dapat mendeteksi warna dan bentuk objek. Dalam pengujian ini dilakukan dengan beberapa parameter yaitu :

- Lampu 15 watt dengan satuan matrix cahaya sebesar 150 lux
- Objek yang dideteksi adalah 3 warna, 3 bentuk yang berbeda dalam 1 gambar
- Sistem akan menggunakan jarak pengambilan gambar sepanjang 5 cm, 10 cm, 20 cm.
- Menggunakan nilai blur (5,5)

Tabel 4.3 Pengujian deteksi Objek berdasarkan Jarak

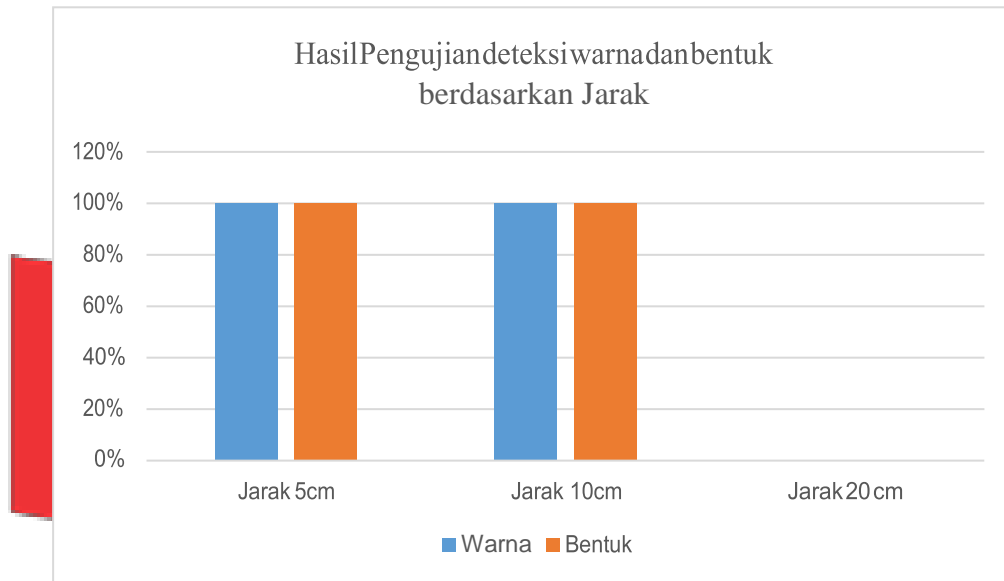
Jarak (cm)	Warna	Bentuk	Hasil Warna	Hasil Bentuk	Presentase	
					Warna	Bentuk
5	Merah	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
	Hijau	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
	Biru	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
10	Merah	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
	Hijau	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
	Biru	△	Terdeteksi	△	100%	100%
		□	Terdeteksi	□	100%	100%
		○	Terdeteksi	○	100%	100%
20	Merah	△	Tidak Terdeteksi	-	-	-
		□	Tidak Terdeteksi	-	-	-
		○	Tidak Terdeteksi	-	-	-
	Hijau	△	Tidak Terdeteksi	-	-	-
		□	Tidak Terdeteksi	-	-	-
		○	Tidak Terdeteksi	-	-	-
	Biru	△	Tidak Terdeteksi	-	-	-
		□	Tidak Terdeteksi	-	-	-
		○	Tidak Terdeteksi	-	-	-

Presentase Keberhasilan deteksi bentuk dari jarak =

$$\frac{\text{Banyaknya bentuk objek yang terdeteksi}}{\text{Banyaknya bentuk objek pada gambar}}$$

Presentase Keberhasilan deteksi warna dari jarak =

$$\frac{\text{Banyaknya warna objek yang terdeteksi}}{\text{Banyaknya warna objek pada gambar}}$$



Gambar 4. 4 Grafik perbandingan Jarak

Dari hasil pengujian diatas didapatkan hasil akhir dari deteksi warna dan bentuk dari jarak 5 cm terdapat nilai keberhasilan mendeteksi warna sebesar 100% dan bentuk sebesar 100%. Untuk jarak 10 cm terdapat nilai keberhasilan mendeteksi warna sebesar 100% dan bentuk sebesar 100%. Pendeteksian dari jarak 20 cm terdapat nilai keberhasilan mendeteksi warna sebesar 0% dan bentuk sebesar 0% karena objek terlalu jauh. Jarak pengambilan yang ideal adalah sebesar 5 cm sampai 10 cm karena presentase sistem pendeteksi warna pada ruangan sebesar 100% dan bentuk 100%.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Adapun kesimpulan yang dapat diambil pada Proyek akhir ini adalah

1. Dapat membuat sistem yang dapat mendeteksi pergerakan objek warna dan bentuk.
2. Dapat membuat sistem yang dapat membedakan antara background dan objek yang akan dideteksi
3. Dapat membuat aplikasi yang dapat membaca warna merah, hijau, biru.
4. Dapat membuat aplikasi yang dapat membaca bentuk seperti, lingkaran, segitiga dan persegi.
5. Sistem pemcahayaan mempengaruhi sistem untuk mendeteksi warna dan bentuk
6. Keberhasilan sistem untuk mendeteksi warna sebesar 100%
7. Keberhasilan sistem untuk medeteksi bentuk sebesar 80%
8. Jarak pengambilan gambar yang ideal adalah 5 cm sampai 10 cm
9. Nilai rata-rata waktu untuk mendeteksi objek pada sistem yaitu 1.67 detik

5.2 Saran

Adapun saran yang dapat diambil pada Proyek Akhir ini adalah

1. Dapat melakukan pengujian dengan objek lain.
2. Dapat membaca warna selain RGB
3. Dapat membaca bentuk selain, lingkaran, segitiga dan persegi
4. Untuk mendapatkan hasil maksimal diharapkan menggunakan resolusi kamera yang lebih tinggi.

Daftar Pustaka

- [1] Astuti. Fajar, *Pengolahan Citra*, Penerbit Andi, Yogyakarta, 2013
- [2] Chandra, Irfan S. *Implementasi Deteksi Warna dengan Orange Pi dan Library OpenCV*, Telkom University, Bandung, 2017
- [3] Eka Adhiyanto, Wiwien Hardikurniawati dan Zuli Budiarso, *Implementasi Metode Image Substracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Bewarna RGB pada file Vide*, 2013
- [4] Jatikusumo, Dwiki. *Panduan Pengolahan Citra*, Universitas Mercu Buana, Jakarta, 2016
- [5] Munir, Rinaldi. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Penerbit Informatika, Bandung, 2004
- [6] OpenCV. 2018. 'RGB-HSV' https://docs.opencv.org/3.3.0/de/d25/imgproc_color_conversions.html ,diakses pada 10 Juni pukul 20.37
- [7] Pradhitya. Rio, *Jurnal Ilmiah, Pembangunan Aplikasi Deteksi dan Tracking Warna Virtual Drawing Menggunakan Algoritma Filtering*, 2014
- [8] Yudana, 2017 <https://www.yudana.id/sqlite-sistem-manajemen-basis-data-berukuran-kecil/> , diakses pada 20 Juni pukul 12.30

