

Design And Construction Of Goods Sorter Conveyor With Color Detection Using ESP-32 Camera Based On Open-CV Python

Ranvang Bangun Konveyor Penyortir Barang Dengan Deteksi Warna Menggunakan ESP-32 Kamera Berbasis Open-CV Python

Hafidz Maulana Ichsan¹⁾, Indah Sulsitiyowati ^{*2)}, Izza Anshory³⁾

¹⁾Program Studi Teknik Elektro, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾ Program Studi Teknik Elektro, Universitas Muhammadiyah Sidoarjo, Indonesia

³⁾ Program Studi Teknik Elektro, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: Indah_sulistiyowati@umsida.ac.id

Abstract. *The use of the OpenCV Python library can be developed in all fields of technology, including in the industrial sector. In the industrial world, there are goods sorting tools in the form of conveyors, where these tools are now increasingly sophisticated with the use of cameras as readers of the objects being sorted. The purpose of applying the OpenCV (Open Source Computer Vision Library) system to this goods sorter conveyor is to make it easier to sort objects based on the color we want and improve education on industrial technology in Indonesia. The method used to select this object is the OpenCV method of detecting objects based on color. The process of detecting this object starts by capturing RGB (red, green, blue) colored objects in real-time, converting RGB colors to HSV, followed by performing a threshold, and after that the morphological process to filter out noise that is not needed in the image by masking the object so that it is centered. The results of this study are to be able to distinguish objects based on RGB color when sorted later by taking into account the value of the HSV on the surface of colored objects, which later this research can be applied to conveyor equipment sorting goods based on color.*

Keywords - OpenCV, Python, Color, Conveyor

Abstrak. *Penggunaan library OpenCV Python dapat dikembangkan di segala bidang teknologi, termasuk di bidang industri. Dalam dunia industri terdapat alat pemilah barang berupa konveyor, dimana alat tersebut kini semakin canggih dengan penggunaan kamera sebagai pembaca objek yang sedang dipilah. Tujuan diterapkannya sistem OpenCV (Open Source Computer Vision Library) pada konveyor pemilah barang ini adalah untuk memudahkan dalam menyortir objek berdasarkan warna yang kita inginkan dan meningkatkan edukasi teknologi industri di Indonesia. Metode yang digunakan untuk menyeleksi objek ini adalah metode OpenCV pendeteksian objek berdasarkan warna. Proses pendeteksian objek ini dimulai dengan menangkap objek berwarna RGB (merah, hijau, biru) secara real-time, mengubah warna RGB menjadi HSV, dilanjutkan dengan melakukan threshold, dan setelah itu proses morfologi menyaring noise yang tidak diperlukan. dalam gambar dengan menutupi objek sehingga berada di tengah. Hasil dari penelitian ini adalah dapat membedakan objek berdasarkan warna RGB saat disortir nantinya dengan memperhatikan nilai HSV pada permukaan objek berwarna, yang nantinya penelitian ini dapat diterapkan pada peralatan conveyor pemilahan barang berdasarkan warna .*

Kata Kunci – OpenCV, Python. Warna, Konveyor

I. PENDAHULUAN

Saat ini kemajuan dan penggunaan AI (Artificial Intelligence) atau teknologi kecerdasan buatan di dunia industri semakin berkembang secara signifikan[1]. Maka perlahan-lahan mereka mulai meninggalkan alat dan mesin tradisional dan beralih ke alat dan mesin modern dengan kontrol otomatis. Klasifikasi atau pemilihan barang khususnya di bidang industri dapat dikelompokkan berdasarkan jenis produk, warna, berat, bentuk, dan lain-lain[2]. Penyortiran dapat dilakukan secara manual oleh manusia, dengan sistem barcode, atau secara otomatis oleh mesin. Penyortiran barang di industri umumnya dilakukan secara manual oleh manusia, sehingga penyortiran barang menjadi lebih lambat, kurang akurat, dan kurang dapat diandalkan karena kelelahan manusia[3], [4]. Klasifikasi atau pemilihan barang pada alat sortir ini menggunakan kamera esp-32 untuk mendeteksi warna barang. Kamera ESP-32 digunakan untuk menangkap gambar barang yang sedang disortir di conveyor. Warna adalah salah satu elemen yang dapat dideteksi dengan baik di dalam kamera sekalipun. Secara khusus, warna yang ditangkap adalah RGB (Red Green Blue)[5].

Sistem pengklasifikasian benda berdasarkan warna dapat dikembangkan dengan berbagai cara. Penelitian oleh Euis W. dkk menggunakan sensor warna TCS230 untuk mendeteksi warna dan mengklasifikasikan barang menggunakan PLC (Programmable Logic Control) sebagai sistem penggeraknya[6],[7]. Warna yang dapat dideteksi adalah merah, hijau, dan biru. Dalam penelitian oleh Ike Sari, dkk. Alat klasifikasi dikembangkan yang dapat

mengklasifikasikan item dengan warna hitam, biru, hijau, merah dan putih. Hasil penelitian menyatakan bahwa sensor warna mendeteksi warna objek pada rentang warna yang ditentukan dan mengaktifkan aktuator tertentu[8],[9]. Dan pada penelitian Wicaksono, F.R., dkk melakukan sortasi barang menggunakan image processing dimana objek yang telah disortir ditangkap oleh kamera kemudian diolah menjadi openCV[10],[11]. Beberapa penelitian tersebut menggunakan metode sensor, Programmable Logic Control), dan image processing. Namun sistem ini dinilai kurang efektif karena tidak menjamin kualitas barang yang disortir setelah diproduksi dan tidak ada media sortir yang lebih baik[12]. Dari kelemahan-kelemahan pada sistem sebelumnya maka dibuatlah sebuah inovasi dengan menggunakan kamera esp-32 sebagai input verifikasi untuk menghitung atau menghitung jumlah barang atau objek yang telah disortir berdasarkan warna dan servo sebagai penyortir juga dapat diamati menggunakan python tampilan software menggunakan metode NumPy dan open-cv yang dapat dipantau secara real time[13]. Dari latar belakang masalah dan sebagian hasil penelitian di atas. Oleh karena itu, penulis merancang konveyor pemilah produk dengan menggunakan dua buah motor servo sebagai pemilah produk berdasarkan tiga kode warna yang berbeda. Dengan menggunakan Arduino UNO R3 sebagai mikrokontroler untuk menggerakkan konveyor dan menyortir barang[14]. Tampilan alat ini menggunakan LCD 16x2 karena penggunaannya yang mudah dipahami dan dipahami semua orang. Sistem kerja alat ini petak dari barang yang sudah lolos produksi akan di sortir menggunakan konveyor ini, barang yang sudah di sortir akan melewati kamera esp-32 dan langsung barang akan di pisahkan sesuai dengan warnanya, tampilan pendeteksi warna bisa diamati menggunakan perangkat lunak tampilan python secara real-time. Ada wadah untuk barang yang sudah disortir berdasarkan warna[15]. Tujuan utama penulis membuat prototipe alat pemilah barang berbasis warna ini adalah untuk membantu dan memudahkan pekerjaan manusia dalam menyortir tiga kode warna secara otomatis dan juga menghitung jumlah barang yang telah disortir[16],[17].

II. METODE

Ada review beberapa jurnal sebelumnya yang sudah menganalisa sistem yang saya rancang. Pada penelitian yang dilakukan oleh Euis W., dkk, pemilahan barang menggunakan sensor warna TCS230 untuk pendeteksian warna menggunakan PLC (Programmable Logic Control) untuk sistem aktuator[6]. Warna yang terdeteksi adalah biru, hijau, dan merah. Dalam penelitian yang dilakukan oleh Ike Sari, dkk, telah dibuat alat klasifikasi yang dapat memilah barang menjadi hitam, putih, merah, biru dan hijau. Hasil menunjukkan bahwa warna objek dapat dideteksi oleh sensor warna dengan rentang warna yang telah ditentukan dan mengaktifkan motor servo tertentu[8]. Dan pada penelitian Wicaksono, F.R., dkk, pemilahan barang menggunakan image processing dimana objek yang telah disortir ditangkap oleh kamera kemudian diolah menjadi openCV[10]. Dari penjelasan di atas maka penulis akan merancang conveyor sortir barang dengan pendeteksi warna menggunakan Arduino UNO R3 sebagai mikrokontroler, kamera ESP-32 sebagai pendeteksi warna barang, LCD I2C 16x2 sebagai tampilan pada alat, motor dc sebagai pengontrol konveyor dan motor servo sebagai penyortir. barang, serta tampilan video real-time yang dapat diamati dengan software python secara real-time.

II.1. Pengolahan Citra

Pengolahan citra adalah suatu proses untuk meningkatkan kualitas, pengambilan, dan perubahan informasi dalam suatu citra. Pengolahan citra telah melalui banyak perkembangan yang dilakukan oleh para peneliti dalam pendeteksian objek. Penelitian ini menunjukkan bahwa warna dapat dijadikan sebagai nilai acuan untuk mendeteksi objek secara langsung melalui kamera[18]. Dalam pengolahan citra, sampling menyatakan ukuran piksel (titik) pada citra, sedangkan kuantisasi menyatakan tingkat kecerahan pada tingkat grayscale sesuai dengan jumlah bit biner yang terdapat pada suatu perangkat optik[19]. Biasanya, ukuran susunan gambar adalah beberapa ratus piksel dikalikan beberapa ratus piksel dan ada beberapa puluh kemungkinan perbedaan tingkat keabuan.

II.2. Open-CV Python

Pada penelitian ini, OpenCV berfungsi sebagai pengolah hasil video untuk objek berwarna merah, hijau, dan biru. Open Source Computer Vision Library (OpenCV) berfungsi untuk mengolah gambar dan video untuk memungkinkan pengguna mengekstrak informasi dari gambar. OpenCV dapat dijalankan dengan menggunakan bahasa pemrograman seperti Python, C, C++, dan Java, serta dapat berjalan di berbagai platform seperti Linux, Windows, Android, Mac OS, dan iOS[20].

II.3. RGB (Red, Green, Blue)

RGB dalam penelitian ini adalah warna objek yang akan dipindahkan yaitu merah, hijau, dan biru (red, green, blue). Setiap warna pada foto/gambar merupakan kombinasi warna RGB (Red, Green, Blue). Pembacaan warna dari 0% hingga 100% adalah 0 hingga 255. RGB (255,0,255), (255,255,0), dan (0,255,255) menghasilkan warna merah, hijau, dan biru[20].

II. 4. HSV (Hue, Saturation, Value)

Pada penelitian ini, HSV berfungsi sebagai pendeteksi warna yang nantinya akan dimasukkan ke dalam Python. Model HSV ini membutuhkan warna primer RGB sebagai dasar pendeteksian warna. H (rona) adalah sudut warna pada sumbu lingkaran kerucut, dengan merah sebagai sumbu 0° . V (nilai) adalah komponen warna sumbu vertikal kerucut. Dan nilai $V=0$ berada di ujung sumbu hitam dan nilai $V=1$ berada di ujung sumbu putih. Sumbu V ini mewakili semua jenis abu-abu. S (saturasi) adalah derajat kejenuhan yang banyak mengandung cahaya putih atau kemurnian warna, dan nilainya adalah radian kerucut[5].

II.5. ESP-32 Camera

Pada penelitian ini kamera Esp-32 berfungsi sebagai input untuk menangkap warna barang yang melewati conveyor. Kamera ESP-32 adalah papan yang berisi WIFI/Bluetooth dengan mikrokontroler ESP-32 dan kamera[21]. Tidak ada USB ke antarmuka serial di papan ini. Pemrograman dilakukan melalui antarmuka eksternal. Umumnya menggunakan USB to Serial FTDI untuk proses pemrograman. Kamera Esp-32 bekerja dengan tegangan 3V dengan pin RX/TX[18].

II.6. Arduino UNO

Pada penelitian ini digunakan board jenis Arduino Uno R3 sebagai mikrokontroler alat ini. Arduino Uno R3 adalah papan mikrokontroler berbasis ATmega328[4]. Spesifikasi Arduino Uno R3 pada tabel 1, adalah sebagai berikut :

Tabel 1. Spesifikasi Arduino UNO [1]

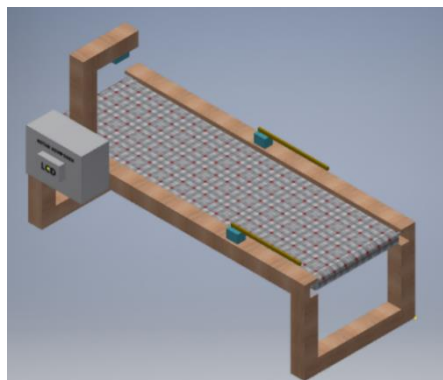
Arduino-UNO	Usage
Mikrokontroler	ATmega328
Number of digital I/O pins	14
Number of analog input pins	6
DC current per I/O pin	40 mA
DC current for the 3.3V pin	50 mA
Input voltage limit	6-20V
operating voltage	5V Recommended input voltage: 7-12V

III. HASIL DAN PEMBAHASAN

III.1. Hasil Pengujian Alat

Pada desain conveyor disesuaikan dengan bentuk balok panjang, namun ukurannya lebih besar karena ditujukan untuk kemasan dengan panjang 1 meter, lebar 40 cm dan tinggi 30 cm. Dengan kamera Esp-32 untuk menangkap gambar objek untuk mendeteksi warna dan menghitung objek yang sedang berjalan pada conveyor yang diletakkan pada conveyor yang diletakkan pada penyangga 40 cm. Arduino uno dan display LCD 16x2 I2C diletakkan di sisi kanan conveyor dalam kotak pelindung agar tidak mudah rusak. Di belakang conveyor terdapat dua motor dc di kedua sisi untuk menggerakkan conveyor. Di bagian depan ditempatkan 2 motor servo dengan jarak 30 cm yang dipisahkan kiri-kanan. Berikut adalah desain kotak paket:

Gambar 1. Desain Alat [1]



Gambar 2. Hasil Alat [2]



III.2. Hasil Program

Pertama-tama buatlah program pada Arduino IDE untuk memprogram koneksi ESP-32 Cam dengan python, berikut program pada Arduino IDE dan program python seperti pada gambar (3) dan (4) :

Gambar 3. Program ESP-32 Camera menggunakan Arduino IDE [3]

```

IP_ESP32_CAMERA $ CONVEYOR_UTAMA LCD_DISPLAY MOTOR_DC_DRIVER_MOTOR SERVO_PENYORTIR
1 # GOODS SORTING CONVEYOR DESIGN PROGRAM WITH COLOR DETECTION USING OPEN-CV PYTHON BASED ESP-32 CAMERA #
2 # ELECTRICAL ENGINEERING THESIS EXAMINATION #
3 # HAFIDZ MAULANA ICHSAN_191020100083 #
4 # COLOR DETECTION AND OBJECT COUNTING PROGRAMS #
5
6 #include <WebServer.h>
7 #include <WiFi.h>
8 #include <esp32cam.h>
9
10 // Enter your wifi ID and Pass
11 const char* WIFI_SSID = "Q?"; //ID wifi
12 const char* WIFI_PASS = "open12345"; //Password wifi
13
14 WebServer server(80);
15
16 // Resolution for ESP-32 Camera
17 static auto loRes = esp32cam::Resolution::find(320, 240); //Low camera resolution
18 static auto miRes = esp32cam::Resolution::find(640, 480); //Medium camera resolution
19 static auto hiRes = esp32cam::Resolution::find(800, 600); //High camera resolution
20 void serveJpg()
21 {
22     auto frame = esp32cam::capture();
23     if (frame == nullptr) {
24         Serial.println("CAPTURE FAIL");
25         server.send(503, "", "");
26         return;
27     }
28     Serial.printf("CAPTURE OK %dxd%d\n", frame->getWidth(), frame->getHeight());
29     static_cast<int>(frame->size());
30 }

```

Gambar 4. Program ESP-32 Camera menggunakan Python [4]

```

# GOODS SORTING CONVEYOR DESIGN PROGRAM WITH COLOR DETECTION USING OPEN-CV PYTHON BASED ESP-32 CAMERA #
# ELECTRICAL ENGINEERING THESIS EXAMINATION #
# HAFIDZ MAULANA ICHSAN_191020100083 #
# COLOR DETECTION AND OBJECT COUNTING PROGRAMS #

import cv2
import urllib.request
import numpy as np
from typing import AsyncIterable
from pyfirmata import Arduino, SERVO, util
import time

def nothing(x):
    pass

# change the IP address below according to the
# IP shown in the Serial monitor of Arduino code
url = 'http://192.168.170.120/cam-hi.jpg'
cv2.namedWindow("Live transmission", cv2.WINDOW_AUTOSIZE)

# Counting Color
counter_green = 0 # Hijau
counter_blue = 0 # Biru
counter_red = 0 # Merah

# Line object detection
Deteksi_Objek = cv2.createBackgroundSubtractorMOG2()
#cek1 = 0
#cek2 = 0

# Arduino port declaration to computer
port = 'COM6'
pin1 = 10
pin2 = 11
board = Arduino(port)

board.digital[pin1].mode = SERVO
board.digital[pin2].mode = SERVO

while(1):

```

Selanjutnya pemrograman untuk menentukan nilai inputan untuk mendapatkan warna yang diinginkan yaitu RGB dengan memasukkan library Python OpenCV seperti pada gambar (5) :

Gambar 5. Program Open-CV menggunakan Python [5]

```

# Green Detection
# Set range for green color and
# Define mask
green_lower = np.array([20, 50, 70], np.uint8)
green_upper = np.array([100, 255, 255], np.uint8)
green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)

# For green color
green_mask = cv2.dilate(green_mask, kernel)
res_green = cv2.bitwise_and(frame40, frame40, mask = green_mask)

# Creating contour to track red color
contours_green, hierarchy_green = cv2.findContours(green_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

for pt, contour in enumerate(contours_green):
    area_green = cv2.contourArea(contour)
    if area_green > 3000:
        x, y, w, h = cv2.boundingRect(contour)
        frame40 = cv2.rectangle(frame40, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame40, "Green Colour", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0))
        if ((x+int(w/2))>int(Widhtb/2)+15)and((x+int(w/2))<(int(Widhtb/2)+30)):
            counter_green = counter_green + 1
        print("Number of Green Objects = ", counter_green)

# REPEAT COUNTING
for cnt_green in contours_green:
    area_green = cv2.contourArea(cnt_green)
    if area_green < 3000 or area_green > 30000:
        continue

# Blue Detection
# Set range for blue color and
# Define mask
blue_lower = np.array([10, 50, 1], np.uint8)
blue_upper = np.array([100, 255, 255], np.uint8)
blue_mask = cv2.inRange(hsvFrame, blue_lower, blue_upper)

```

Selanjutnya program sorting menggunakan 2 buah motor servo pada program python yang dijelaskan pada gambar (6) sebagai berikut :

Gambar 6. Pemrograman Penyortiran Motor Servo menggunakan Python [6]

```

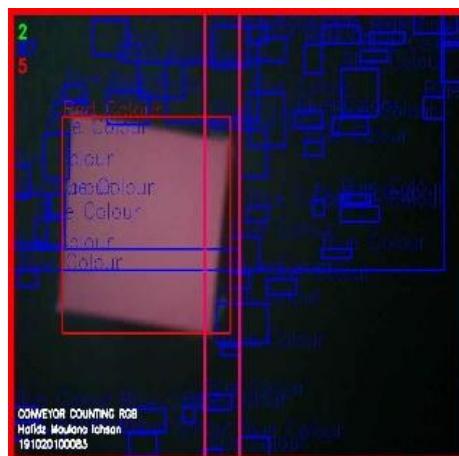
def rotateservo(pin1, angle):
    board.digital[pin1].write (angle)
def rotateservo(pin2, angle):
    board.digital[pin2].write (angle)

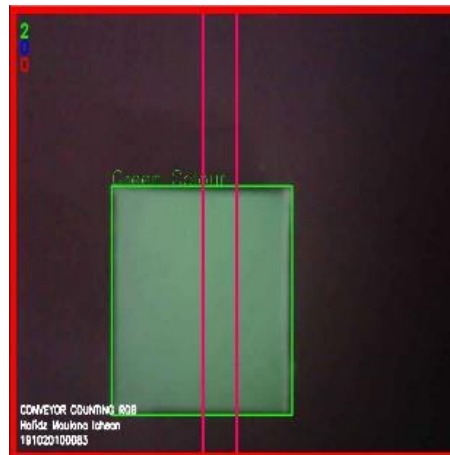
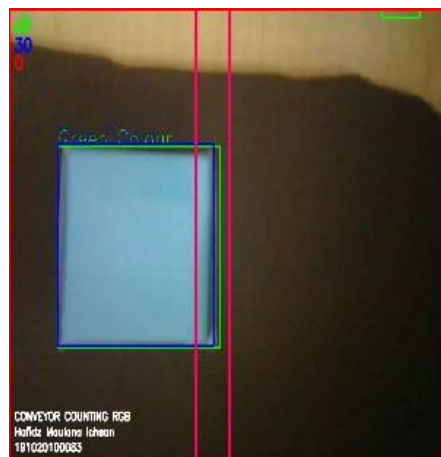
while True:
    x = input("input : ")
    if x=="1":
        for i in range(0,45):
            rotateservo(pin1,i)
    elif x=="2":
        for i in range(0,10):
            rotateservo(pin1,i)
    if x=="3":
        for i in range(0,45):
            rotateservo(pin2,i)
    elif x=="4":
        for i in range(0,10):
            rotateservo(pin2,i)

```

III.3. Hasil Pengujian Pembacaan ESP-32 untuk Warna RGB

Dalam pengujian Kamera ESP-32 sebagai pembacaan warna RGB dilakukan dengan menggunakan software Python dan Arduino IDE, pengujian dilakukan untuk mendapatkan data citra warna RGB yang diinginkan, berikut hasil pembacaan warna RGB yang ditunjukkan pada angka 7, 8, 9 :

Gambar 7. Warna Merah [7]

Gambar 8. Warna Hijau [8]**Gambar 9.** Warna Biru [9]

Dari gambar 7, 8, 9 diperoleh hasil akurasi pembacaan setiap warna RGB dari jarak 12cm yang dijelaskan pada tabel 2 sebagai berikut :

Tabel 2. Hasil Pengujian Pembacaan Warna RGB menggunakan ESP-32 Camera [2]

RGB Color Reading Results					
	HSV value			Distance (cm)	Accuracy (%)
	H (Hue)	S (Saturation)	V (Value)		
R	Lower : 136 Upper : 180	Lower : 87 Upper : 255	Lower : 111 Upper : 255	12 cm	80%
G	Lower : 25 Upper : 102	Lower : 52 Upper : 255	Lower : 72 Upper : 255	12 cm	100%
B	Lower : 94 Upper : 120	Lower : 80 Upper : 255	Lower : 2 Upper : 255	12 cm	80%

Tampilan di software python menunjukkan bahwa Cam ESP32 sangat rentan terhadap pembacaan cahaya, sehingga sering terjadi noise saat membaca warna.

III.4. Hasil Pengujian Perhitungan Saat Penyortiran

Pada pengujian ini untuk perhitungan item yang akan diurutkan. Hasil perhitungan item yang diurutkan menunjukkan ketepatan menangkap kesesuaian setiap 10 objek sesuai dengan warna RGB, yang dijelaskan pada tabel 3 sebagai berikut.

Tabel 3. Hasil Pengujian Perhitungan Penyortiran Barang [3]

Counted number of colored objects		
Color	Number of colored objects (10 obejct)	Accuracy (%)
Red	8	80%
Green	10	100%
Blue	8	80%

Dari percobaan untuk mengetahui akurasi Cam ESP32 dalam menghitung setiap warna RGB, didapatkan bahwa menurut gambar 13 diatas, warna merah memiliki akurasi sebesar 80%, warna hijau memiliki akurasi 100%, dan warna biru memiliki akurasi sebesar 80%.

III.5. Hasil Pengujian Tingkat Keakurasian Barang Yang disortir

Pada pengujian ini untuk mengukur ketelitian penyortiran barang yang akan disortir. Hasil penyortiran barang menunjukkan ketepatan barang yang disortir setiap 10 kali, dijelaskan pada tabel 4 sebagai berikut.

Tabel 4. Hasil Pengujian Tingkat Keakurasian Barang yang disortir [4]

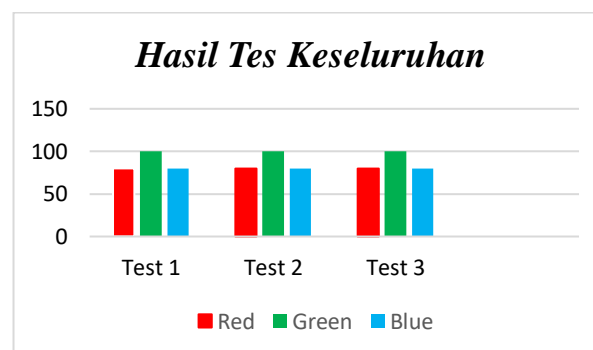
Color	testing	Sorter 1	Sorter 2	Sorter 3	Accuracy (%)
Red	10 times	8	-	-	80%
Green	10 times	-	10	-	100%
Blue	10 times	-	-	8	80%

Dari percobaan untuk mengetahui akurasi motor servo dalam menyortir objek warna RGB yang dilakukan sebanyak 10 kali, didapatkan bahwa sesuai tabel 3 diatas, warna merah memiliki akurasi 80%, warna hijau 100% , warna biru adalah 80%.

III.6. Hasil Keseluruhan Pengujian

Hasil perhitungan objek yang disortir menunjukkan ketelitian yang tepat, sesuai dengan warna yang diinginkan. Dari ketiga hasil pengujian di atas diperoleh grafik hasil penelitian secara keseluruhan yang dijelaskan pada gambar 10 sebagai berikut:

Gambar 10. Hasil Keseluruhan Pengujian [10]



V. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, dapat diperoleh beberapa kesimpulan sebagai berikut.

1. Pengolahan citra dengan metode OpenCV pada Kamera ESP-32 dapat berjalan dengan baik, dengan 3 kategori warna yaitu: merah, hijau, biru.
2. Program python yang telah dibuat dapat berfungsi dengan baik dan dapat bekerja sesuai dengan yang dirancang.
3. Dengan metode OpenCV untuk membaca 3 warna ini dapat dengan mudah memisahkan objek merah, hijau, biru.
4. Ada kekurangan pada Cam ESP32 yang sangat rentan terhadap cahaya sehingga menimbulkan gangguan pada saat proses pembacaan.
5. Esp-32 kamera juga memiliki kelemahan terhadap suhu panas pada perangkat, yang akan menyebabkan keterlambatan pada proses pembacaan.

Pada penelitian selanjutnya ditambahkan kamera yang digunakan agar pengenalan lebih presisi dan pola enumerasi diperbanyak untuk meminimalisir pengaruh bayangan yang mengganggu proses pengenalan citra. Perlu mengontrol kecepatan motor konveyor, agar pemilihan objek bisa lebih baik.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam penulisan penelitian ini, khususnya kepada program studi teknik elektro di Universitas Muhammadiyah Sidoarjo.

REFERENSI

- [1] S. H. Park and K. Han, "Methodologic Guide for Evaluating Clinical Performance and Effect of Artificial Intelligence Technology for Medical Diagnosis and Prediction 1 REVIEW: Evaluation of Artificial Intelligence Tools for Diagnostic or Predictive Analysis Park and Han," *Radiol. n Radiol. Radiol.*, vol. 286, no. 3—March, 2018.
- [2] D. M. Rajaguguk and I. Panjaitan, "Compression of Color Image Using Quantization Method," vol. 10, no. 02, pp. 13–18, 2020.
- [3] F. Jalled and I. Voronkov, "Object Detection using Image Processing," pp. 1–6, 2016, [Online]. Available: <http://arxiv.org/abs/1611.07791>
- [4] S. Suwarno and K. Kevin, "Analysis of Face Recognition Algorithm: Dlib and OpenCV," *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 173–184, 2020, doi: 10.31289/jite.v4i1.3865.
- [5] X. Li, S. Li, Q. Liu, Z. Cui, and Z. Chen, "A Triple-Channel Colorimetric Sensor Array for Identification of Biothiols Based on Color RGB (Red/Green/Blue) as Signal Readout," *ACS Sustain. Chem. Eng.*, vol. 7, no. 20, pp. 17482–17490, 2019, doi: 10.1021/acsschemeng.9b04740.
- [6] H. Soleh, E. W. H. Witarsa, M. Ferdian, D. Yuniarti, and C. Caroline, "Prototipe Penyortir Barang Berdasarkan Warna, Bentuk Dan Tinggi Berbasis Programmable Logic Controller (Plc) Dengan Penggerak Sistem Pneumatic," *J. Mikrotiga*, vol. 1, no. 2, 2014.
- [7] T. J. Nuva, M. I. Ahmed, and S. S. Mahmud, "Design & Fabrication of Automatic Color & Weight-Based Sorting System on Conveyor Belt," *J. Integr. Adv. Eng.*, vol. 2, no. 2, pp. 147–157, 2022, doi: 10.51662/jiae.v2i2.87.
- [8] M. I. Sari, R. Handayani, S. Siregar, and B. Isnu, "Pemilah Benda Berdasarkan Warna Menggunakan Sensor Warna TCS3200," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 4, no. 2, pp. 85–90, 2018, doi: 10.15575/telka.v4n2.85-90.
- [9] M. Kamalakannan and K. Devadharshini, "Controlling the Speed of Conveyor Belt using Python – Raspberry Pi 3B+," *Orient. J. Comput. Sci. Technol.*, vol. 12, no. 2, pp. 57–64, 2019, doi: 10.13005/ojcest12.02.05.
- [10] F. R. Wicaksono *et al.*, "Perancangan Dan Implementasi Alat Penyortir Barang Pada Design and Implementation of Items Device Sorting on Conveyor," *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 5, no. 1, pp. 40–47, 2018.
- [11] A. Thihe, Z. Z. Moe San, and D. Z. Min Oo, "Design and Development of an Automatic Color Sorting Machine on Belt Conveyor," *Int. J. Sci. Eng. Appl.*, vol. 8, no. 7, pp. 176–179, 2019, doi: 10.7753/ijsea0807.1002.
- [12] P. Liptai, E. Lumnitzer, M. Moravec, and M. Piňosová, "Analysis and Classification of Noise Sources of Conveyor Systems by Sound Visualizing on the Postal Package Sorting Line," *Adv. Sci. Technol. Res. J.*, vol. 12, no. 4, pp. 172–176, 2018, doi: 10.12913/22998624/100348.
- [13] M. D. Cookson and P. M. R. Stirk, "ZOOMORPHIC MOBILE ROBOT DEVELOPMENT FOR VERTICAL MOVEMENT BASED ON ESP 32- CAM Vladyslav," 2019.
- [14] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID system regulator for mini conveyor drive based-on matlab," *J. Robot. Control*, vol. 1, no. 6, pp. 185–190, 2020, doi: 10.18196/jrc.1636.
- [15] A. Mordvintsev and K. Abid, "OpenCV-Python Tutorials Documentation," *OpenCV Python Doc.*, p. 269, 2017, [Online]. Available: <https://media.readthedocs.org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals.pdf>
- [16] A. M. Elhanafi, R. Siregar, M. Yeni, and S. An-nissa, "Cryptography Application on RGB Overlapping Block Based PVD Using AES," vol. 7, no. 3, pp. 2116–2124, 2022.
- [17] D. Sonny Febriyan and R. Dwi Puriyanto, "Implementation of DC Motor PID Control On Conveyor for Separating Potato Seeds by Weight," *Int. J. Robot. Control Syst.*, vol. 1, no. 1, pp. 15–26, 2021, doi: 10.31763/ijrcs.v1i1.221.
- [18] A. Shukla and R. Diwan, "Iot based load automation with remote access surveillance using esp 32 camand esp 8266 module," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 3, pp. 6904–6914, 2021.
- [19] N. M. Syahrian, P. Risma, and T. Dewi, "Vision-Based Pipe Monitoring Robot for Crack Detection Using Canny Edge Detection Method as an Image Processing Technique," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 2, no. 4, pp. 243–250, 2017, doi: 10.22219/kinetik.v2i4.243.
- [20] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: An open source differentiable computer vision library for PyTorch," *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 3663–3672, 2020, doi: 10.1109/WACV45572.2020.9093363.
- [21] S. C. S. Yanti and I. Sulistiyowati, "An Inventory Tool for Receiving Practicum Report Based on IoT by Using ESP32-CAM and UV Sterilizer: A Case Study at Muhammadiyah University of Sidoarjo," *J. Electr. Technol. UMY*, vol. 6, no. 1, pp. 49–56, 2022, doi: 10.18196/jet.v6i1.14607.