



**Sistem Sortir Barang Berdasarkan Warna dan
Bentuk Menggunakan *Webcam* Berbasis *PLC*
*Outseal***

Tugas Akhir

Oleh:

Shendy Saputra (42121201025)

Anastasya Andaresta Pelmelay (4212111008)

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2024**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul: "Sistem Sortir Barang Berdasarkan Warna dan Bentuk Menggunakan *Webcam* Berbasis *PLC Outseal*" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 21 Januari 2025



Anastasya Andaresta Pelmelay
NIM: 4212111008



Shendy Saputra
NIM: 4212101025

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr. T)
di
Politeknik Negeri Batam

Oleh:
Shendy Saputra (4212101025)
Anastasya Andaresta Pelmelay (4212111008)

Tanggal Sidang: 21 01, 2025

Disetujui oleh:



1. Dr. Ir Budi Sugandi S.T., M.Eng.,
IPM
NIK: 100002



1. Ir. Daniel Sutopo Pamungkas
S.T., M.T., Ph. D
NIK: 100006



2. M. Naufal Airlangga Diputra S.Pd.,
M.P.H
NIK: 122281

Sistem Sortir Barang Berdasarkan Warna dan Bentuk Menggunakan *Webcam* Berbasis *PLC Outseal*

Abstrak

Sistem ini memanfaatkan webcam sebagai sensor untuk mendeteksi objek berdasarkan warna dan bentuk, sementara kendali utama menggunakan *PLC Outseal*. Dengan menggunakan algoritma *YOLO v5*, yang dikenal efektif dalam deteksi objek, sistem ini mampu mengidentifikasi dan mengklasifikasikan objek secara *real-time*. Pemrograman menggunakan *Ladder Diagram* memungkinkan kontrol yang efisien terhadap perangkat keras *PLC*. Penggunaan pencahayaan yang optimal juga merupakan hal yang penting untuk dipertimbangkan dalam melakukan pengujian pendeteksian objek. Dalam pengujian, sistem menunjukkan hasil akurasi 88,26% dan presisi 70,38%.

Kata kunci: *PLC Outseal*; Sistem Automasi; Sistem Sortir.

Sorting System Based on Color and Shape using Outseal PLC based Webcam

Abstract

This system uses a webcam as a sensor to detect objects based on color and shape, while the main control uses an Outseal PLC. By using the YOLO v5 algorithm, which is known to be effective in object detection, this system is able to identify and classify objects in real-time. Programming using Ladder Diagrams allows efficient control of PLC hardware. Optimal use of lighting is also an important thing to consider when conducting object detection testing. In testing, the system showed accuracy 88,26% and precision 70,38%.

Keywords: Automation Systems; Outseal PLC; Sorting System.

Kata Pengantar

Alhamdulillahirabbil'alamin. Puji syukur ke hadirat Allah SWT atas limpahan rahmat dan hidayah-Nya. Shalawat dan salam semoga tercurahkan kepada nabi besar Rasulullah Muhammad SAW. Sehingga penulis mampu menyelesaikan Laporan Tugas Akhir yang berjudul "Sistem Sortir Barang Berdasarkan Warna dan Bentuk Menggunakan *Webcam* Berbasis *PLC Outseal*". Penulisan Laporan Tugas Akhir ini memiliki tujuan untuk melengkapi persyaratan kelulusan tingkat Diploma IV Program Studi Mekatronika Politeknik Negeri Batam.

Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya atas segala bantuan dan dukungan yang telah diberikan selama penyusunan Tugas Akhir ini hingga selesai kepada:

1. Kedua orang tua dan keluarga yang telah memberikan doa, semangat dan motivasi.
2. Bapak Ir. Bambang Hendrawan, ST., MSM., CIPMP., CISCIP. selaku Direktur Politeknik Negeri Batam.
3. Bapak Dr. Budi Sugandi, S.T., M.Eng. selaku ketua Jurusan Teknik Elektro Politeknik Negeri Batam (2016-2024).
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng. selaku ketua Jurusan Teknik Elektro Politeknik Negeri Batam (2025-2029).
5. Bapak Indra Hardian Mulyadi, S.T., M.Eng. selaku ketua Program Studi Mekatronika Politeknik Negeri Batam (2020-2024).
6. Bapak Diono S.Tr. T., M.Sc. selaku ketua Program Studi Mekatronika Politeknik Negeri Batam (2025-2029).
7. Bapak Ir. Daniel Sutopo Pamungkas S.T., M.T., Ph. D selaku dosen pembimbing yang telah menyediakan waktu dan pikiran untuk mengarahkan penulis dalam menyelesaikan Tugas Akhir.
8. Ibu Nadhrah Wivanus S.Si., M. Si selaku dosen pengampu mata kuliah Tugas Akhir.
9. Bapak Diono S.Tr. T., M.Sc. selaku wali dosen kelas A malam.
10. Bapak Dr. Budi Sugandi, S.T., M.Eng. dan Bapak M. Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen penguji.
11. Seluruh dosen-dosen Jurusan Teknik Elektro Politeknik Negeri Batam.
12. Seluruh teman-teman yang telah membantu proses penyelesaian Tugas Akhir ini.

Penulis menyadari dalam penyusunan Laporan Tugas Akhir ini masih banyak kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Tugas Akhir ini. Terima kasih atas segala perhatiannya.

Batam, 21 Januari 2025

Shendy Saputra
Anastasya Andaresta Pelmelay

Daftar Isi

Pernyataan Keaslian Tugas Akhir	i
Lembar Pengesahan	ii
Abstrak	iii
<i>Abstract</i>	iv
Kata Pengantar	v
Daftar Isi	vii
Daftar Gambar	x
Daftar Tabel	xi
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Manfaat	2
1.5. Batasan	3
1.6. <i>Work Breakdown Structure</i>	3
Bab 2. Tinjauan Pustaka	4
2.1. Konveyor	4
2.2. <i>Stepper Motor NEMA23</i>	4
2.3. <i>Microstep Driver TB6600</i>	5
2.4. <i>Sensor Proximity</i>	6
2.5. <i>Webcam</i>	7
2.6. <i>PLC Outseal</i>	7
2.7. <i>Yolo V5</i>	8
2.8. <i>Confusion Matrix</i>	9

2.8.1. Akurasi	9
2.8.2. Presisi	9
Bab 3. Metodologi Penelitian	10
3.1. Perancangan	10
3.1.1. Perancangan Sistem Kerja	11
3.1.2. Perancangan Sistem Mekanik	12
3.1.3. Perancangan Sistem Elektrikal	13
3.1.4. Perancangan Sistem Pendeteksi Bentuk dan Warna	15
3.2. Alat dan Bahan.....	16
3.3. Pengujian	18
3.3.1. Pengujian Deteksi Adanya Benda	21
3.3.2. Pengujian Deteksi Bentuk Dan Warna	21
Bab 4. Hasil dan Pembahasan.....	23
4.1. Hasil yang Telah Dicapai	23
4.1.1. Alat Pemilah Benda	23
4.1.2. Kelistrikan Pada Alat Pemilah Benda	24
4.1.3. Hasil Pengujian Deteksi Adanya Benda	25
4.1.4. Pengujian Deteksi Bentuk Dan Warna	29
4.1.5. Pengujian <i>Sequence</i> (Proses) Pada Alat Pemilah Benda	44
4.2. Pembahasan	46
4.2.1. Pembahasan Hasil Pengujian Deteksi Adanya Benda	46
4.2.2. Pembahasan Hasil Pengujian Deteksi Bentuk dan Warna.....	47
Bab 5. Kesimpulan dan Saran	56
5.1. Kesimpulan.....	56

5.2. Saran	56
Daftar Pustaka	57
Biodata	59
Lampiran	60

Daftar Gambar

Gambar 1. <i>Conveyor Belt</i>	4
Gambar 2. <i>Stepper Motor NEMA23</i>	5
Gambar 3. <i>Microstep Driver TB6600</i>	5
Gambar 4. <i>Sensor Proximity</i>	6
Gambar 5. <i>WebCam</i>	7
Gambar 6. <i>PLC Outseal</i>	7
Gambar 7. Diagram Alur Pengerjaan	10
Gambar 8. Diagram Alur Sistem Kerja	11
Gambar 9. <i>Design Alat</i>	12
Gambar 10. <i>Schematic Wiring PLC Outseal</i>	13
Gambar 11. <i>Schematic Wiring Arduino Uno</i>	14
Gambar 12. Diagram Alur Sistem Pendeteksi Bentuk dan Warna	15
Gambar 13. Program <i>Phyton</i> Pendeteksi Objek	21
Gambar 14. Program <i>Phyton</i> Pendeteksi Bentuk dan Warna	22
Gambar 15. Hasil dari Perancangan Mekanikal	23
Gambar 16. Hasil dari Perancangan Elektrikal	24
Gambar 17. Hasil dari Pengujian Deteksi Benda dan Pergerakan <i>Servo 1</i>	46
Gambar 18. Hasil dari Pengujian Deteksi Benda dan Pergerakan <i>Servo 2</i>	46
Gambar 19. Hasil dari Pengujian Deteksi Benda dan Pergerakan <i>Servo 3</i>	47
Gambar 20. Model Klasifikasi <i>Confusion Matrix</i> pada Hasil Pengujian Deteksi Bentuk dan Warna menggunakan <i>Ringlight</i>	48
Gambar 21. Model Klasifikasi <i>Confusion Matrix</i> pada Hasil Pengujian Deteksi Bentuk dan Warna pada Siang Hari	50
Gambar 22. Model Klasifikasi <i>Confusion Matrix</i> pada Hasil Pengujian Deteksi Bentuk dan Warna pada Sore Hari	52

Daftar Tabel

Tabel 1. <i>WBS</i>	3
Tabel 2. Estimasi Biaya	16
Tabel 3. Klasifikasi Benda yang Diuji.....	18
Tabel 4. Hasil Deteksi Adanya Benda	25
Tabel 5. Hasil Deteksi Bentuk dan Warna dengan <i>Ringlight</i>	30
Tabel 6. Hasil Deteksi Bentuk dan Warna pada Siang Hari.....	35
Tabel 7. Hasil Deteksi Bentuk dan Warna pada Sore Hari	40
Tabel 8. Hasil Pengujian <i>Sequence</i> (Proses) Pada Alat Pemilah Benda	45
Tabel 9. Hasil Klasifikasi Deteksi Bentuk dan Warna menggunakan <i>Ringlight</i>	48
Tabel 10. Hasil Klasifikasi Deteksi Bentuk dan Warna pada Siang Hari.....	50
Tabel 11. Hasil Klasifikasi Deteksi Bentuk dan Warna pada Sore Hari.....	52
Tabel 12. Hasil Perhitungan Akurasi dan Presisi	54

Bab 1. Pendahuluan

1.1. Latar Belakang

Saat ini, industri semakin mendorong pengembangan sistem otomatisasi untuk meningkatkan efisiensi dalam proses produksi dan penyortiran barang. Dalam konteks ini, penting untuk memilah barang dengan tepat berdasarkan kriteria tertentu seperti warna dan bentuk untuk memastikan kualitas produk akhir serta meminimalkan cacat atau kesalahan dalam produksi. Penyortiran barang dapat dilakukan dengan mengelompokkan jenis, warna atau bentuk barang. Penyortiran dapat dilakukan secara manual menggunakan tenaga manusia, namun penyortiran bahan yang secara manual dimana akan membutuhkan waktu yang lama, memerlukan biaya upah tenaga kerja dan rentan terjadi kesalahan yang dilakukan oleh manusia karena sifat yang mudah bosan mengingat warna bahan yang akan di sortir memiliki warna yang hampir sama secara keseluruhan [1]. Dukungan dari kemajuan teknologi penglihatan komputer memungkinkan penggunaan kamera dan teknologi pengolahan citra yang semakin canggih dan terjangkau. Perancangan konveyor yang akan dibuat dapat mengenali pola tertentu yakni menggunakan pola bentuk lingkaran, dan pola persegi atau bukan keduanya [2]. Saat ini sedang berkembang suatu modul perpaduan antara PLC dengan Mikrokontroler. Produk tersebut merupakan karya anak bangsa dengan produk *PLC Outseal*. *PLC Outseal* merupakan teknologi otomasi karya anak bangsa. *PLC Outseal* merupakan perangkat keras layaknya PLC pada umumnya yang digunakan untuk merancang kontrol otomasi industri. Basis *PLC Outseal* adalah *Arduino Uno* dengan bahasa pemrograman *ladder diagram*. Dibutuhkan perangkat lunak yang bernama *Outseal Studio* yang juga merupakan produk dari *Outseal*. *Outseal Studio* dijalankan di PC dalam bentuk *visual programming* menggunakan *ladder diagram* [3]. Penggunaan *PLC Outseal*, sebuah jenis PLC yang sering digunakan pengolahan sinyal, menjadi solusi yang cocok untuk mengontrol sistem pemrosesan gambar [4]. Dengan menggunakan webcam sebagai sensor penglihatan, sistem ini dapat diimplementasikan dengan biaya yang relatif rendah, ketersediaan yang luas, dan kemudahan penggunaan [5]. Dengan demikian, pengembangan sistem sortir barang berbasis *webcam* menggunakan *PLC Outseal* merupakan suatu pilihan yang signifikan dalam memenuhi kebutuhan industri akan otomatisasi proses pemilahan barang secara efisien, ekonomis dan efektif [4]. Industri semakin beralih ke otomatisasi untuk meningkatkan proses produksi dan pengelolaan barang. Dalam konteks ini, pemilahan barang berdasarkan kriteria tertentu seperti warna dan bentuk merupakan tahap penting untuk memastikan kualitas produk akhir dan meminimalkan cacat atau kesalahan dalam 1 produksi. Warna tersebut dibedakan menjadi tiga yaitu warna primer, warna sekunder dan warna tersier [5]. Permasalahan umum yang dihadapi dalam bidang ini adalah kebutuhan akan sistem pemilahan barang yang dapat menyortir barang

berdasarkan dua aspek yaitu warna dan bentuk. Solusi yang telah diimplementasikan oleh penelitian sebelumnya meliputi penggunaan sensor warna *TCS230* yang dipergunakan untuk mengidentifikasi warna sampel kandungan penggunaan formalin pada ikan asin [6], untuk menentukan komposisi warna cat yang akan digunakan untuk mengecat mobil [7], cara mengukur regenerasi warna pada sensor warna dengan metode kecerdasan buatan [8]. Beberapa produk sejenis yang telah ada di pasaran adalah sistem sortir barang berbasis penglihatan komputer yang menggunakan kamera dan *PLC*. Namun, evaluasi terhadap produk-produk tersebut menunjukkan adanya *gap* yang dapat dieksplorasi lebih lanjut, baik dari segi teknis maupun non-teknis. Misalnya ergonomi, keakuratan, harga, dan kemampuan dalam mendeteksi objek yang terbatas berupa bentuk saja ataupun warna saja, menjadi pertimbangan penting dalam merumuskan solusi yang optimal untuk pemilahan barang berdasarkan warna dan bentuk menggunakan webcam berbasis *PLC Outseal*.

1.2. Rumusan Masalah

Dari uraian latar belakang diatas maka perumusan masalahnya adalah sebagai berikut:

1. Bagaimana cara menganalisis sebuah sistem sortir barang berdasarkan warna dan bentuk yang menggunakan kamera dan perangkat kontrol *PLC Outseal*?
2. Bagaimana cara menganalisis perbedaan bentuk dan warna sebuah barang menggunakan sistem sortir barang yang menggunakan kamera dan perangkat kontrol *PLC Outseal*?
3. Bagaimana cara menganalisis bentuk dan warna sebuah barang pada *webcam* berbasis *PLC Outseal* untuk mengetahui keakuratan terkait bentuk dan warna?

1.3. Tujuan

Berdasarkan rumusan masalah yang ada, dapat disimpulkan tujuan tugas akhir ini adalah sebagai berikut:

1. Menganalisis alat yang dapat memilah barang berdasarkan warna dan bentuk dengan menggunakan kamera dan perangkat kontrol *PLC Outseal*.
2. Menganalisis sistem sortir barang yang mampu mengenali bentuk dan warna objek meskipun memiliki gradasi warna yang menyerupai.
3. Menganalisis bentuk dan warna sebuah barang pada *webcam* berbasis *PLC Outseal* untuk mengetahui keakuratan terkait bentuk dan warna.

1.4. Manfaat

Sistem ini diharapkan dapat digunakan sebagai alat bantu yang memudahkan pekerjaan manusia khususnya dalam pemilahan barang dan juga sebagai penelitian dalam pengenalan benda menggunakan *webcam* dan *PLC Outseal*.

1.5. Batasan

Penulis menetapkan beberapa batasan masalah pada perancangan ini, yaitu:

1. Masukan terdiri dari tiga bentuk, yaitu tabung, kotak, dan bintang, serta enam warna, yaitu merah, merah salmon, putih, abu-abu, biru tua dan biru muda dengan diameter benda 4 cm dan tinggi 2.5 cm
2. Dalam sistem kamera hanya mampu mendeteksi satu barang dalam satu *frame*.
3. Tinggi dari penempatan kamera dengan barang yang dideteksi adalah 17 cm.
4. Pencahayaan ditempatkan di atas kamera untuk mendukung proses deteksi.

1.6. Work Breakdown Structure

Tabel 1. WBS

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Shendy Saputra	Koordinator, mekanikal, elektrikal, pengujian, pemrograman <i>Phyton</i> dan Pemrograman <i>C</i>
2	Anastasya Andaresta Pelmelay	Pemrograman <i>Outseal</i> , pengumpulan data dan analisis statistik

Bab 2. Tinjauan Pustaka

2.1. Konveyor



Gambar 1. *Conveyor Belt*

Konveyor adalah bagian umum dari peralatan penanganan material mekanis yang bergerak dari satu lokasi ke lokasi lain. Konveyor terutama berguna dalam aplikasi yang melibatkan transportasi bahan berat atau besar. Sistem konveyor memungkinkan transportasi cepat dan efisien untuk berbagai bahan. Pemilihan konveyor untuk pengangkutan material padatan antara lain tergantung pada:

1. Kapasitas material yang ditangani.
2. Jarak perpindahan material.
3. Kondisi pengangkutan: horizontal, vertikal atau inklinasi.
4. Ukuran (*size*), bentuk (*shape*) dan sifat material (*properties*).
5. Harga peralatan tersebut.

Konveyor *belt* pada dasarnya merupakan peralatan yang cukup sederhana. Alat tersebut terdiri dari sabuk yang tahan terhadap pengangkutan benda padat. Sabuk yang digunakan pada konveyor *belt* ini dapat dibuat dari berbagai jenis bahan misalnya dari karet, plastik, kulit ataupun logam yang tergantung dari jenis dan sifat bahan yang akan diangkut. Untuk mengangkut bahan-bahan yang panas, sabuk yang digunakan terbuat dari logam yang tahan terhadap panas [9].

2.2. Stepper Motor NEMA23

Stepper Motor DC NEMA23 adalah jenis motor *stepper* yang memiliki ukuran *frame* 2.3 x 2.3 inci (atau sekitar 58.4 x 58.5 mm) dan sudut langkah sebesar 1.8°, yang berarti motor ini dapat melakukan 200 langkah per revolusi. Setiap fase motor ini menarik arus sebesar 2.8 A pada 3.2 V, yang memungkinkan motor untuk memiliki torsi pegangan sebesar 19 kg-cm. Motor ini umumnya digunakan dalam *printer*, mesin *CNC*, aktuator linier, dan *hard drive* karena kemampuannya untuk mengontrol posisi dan kecepatan dengan presisi tinggi [11].

Motor *stepper* beroperasi dengan menerima *input* berupa pulsa-pulsa digital, berbeda dengan motor *DC* konvensional yang bekerja berdasarkan komutasi pada komponen *brush* (sikat) nya. Motor ini bergerak dalam langkah-langkah diskrit, yang memungkinkan kontrol gerakan yang sangat akurat. Kecepatan gerakan motor *stepper* diukur dalam *step* per detik, dan konstruksi motor *stepper* terdiri dari *rotor* dan *stator*, namun tidak memiliki sikat karena putarannya dikontrol dengan memberikan pulsa diskrit pada kumparan motor [12].



Gambar 2. Stepper Motor NEMA23

2.3. Microstep Driver TB6600

Microstep Driver TB6600 adalah modul *driver* motor *stepper* yang menyediakan kemampuan *microstepping* untuk motor *stepper*. *Driver* ini dibangun di sekitar IC *Toshiba TB6600HG* dan dapat digunakan untuk mengendalikan motor *stepper* bipolar dua-fase. Dengan arus maksimum 3.5 A secara berkelanjutan, *driver TB6600* dapat digunakan untuk mengontrol motor *stepper* yang cukup besar seperti *NEMA23* [13].



Gambar 3. Microstep Driver TB6600

Driver ini mendukung kontrol kecepatan dan arah, dan Anda dapat mengatur *microstep* dan arus keluarannya dengan 6 saklar *DIP*. Pengaturan *microstep* memungkinkan motor untuk bergerak dalam langkah-langkah yang lebih kecil dari langkah penuh, yang menghasilkan gerakan yang lebih halus dan presisi yang lebih tinggi. Misalnya, jika motor *stepper* memerlukan 200 pulsa untuk satu putaran penuh, dengan *microstepping*, Anda dapat membagi setiap langkah menjadi bagian yang lebih kecil, seperti 1/2, 1/4, 1/8, 1/16, dan bahkan hingga 1/32 atau lebih, tergantung pada *driver* [14].

TB6600 juga dilengkapi dengan pengaturan arus yang dapat disesuaikan, yang memungkinkan Anda untuk mengatur arus yang tepat untuk motor *stepper* Anda, sehingga mencegah *overheating* dan meningkatkan efisiensi operasional. *Driver* ini cocok untuk digunakan dengan mikrokontroler apa pun yang menyediakan sinyal 5V, seperti *Arduino*, dan dapat digunakan dalam berbagai aplikasi seperti mesin *CNC*, *printer 3D*, dan sistem otomasi lainnya [15].

2.4. Sensor *Proximity*



Gambar 4. Sensor *Proximity*

Sensor *proximity* adalah alat atau perangkat yang dapat mendeteksi perubahan jarak pada suatu benda. Namun proses tersebut terjadi dengan tanpa adanya kontak fisik. sensor jarak di Indonesia juga familiar dengan istilah sensor jarak. Dalam prosesnya, sensor jarak memakai pengantar radiasi elektromagnetik. Inilah yang membuat perangkat bisa mendeteksi keberadaan benda atau kondisinya meskipun tanpa ada kontak fisik. Contoh pemanfaatan sensor jarak sering digunakan untuk kepentingan yang sangat beragam. Diantaranya ada yang digunakan untuk mendeteksi bahan. Selain itu, ada pula yang digunakan untuk mendeteksi lingkungan yang berbeda. Pengaplikasiannya yaitu seperti digunakan pada *smartphone* ataupun berbagai perangkat elektronik yang lainnya. Sensor *proximity* juga sering digunakan untuk beragam mesin industri. Contohnya seperti mesin plastik, mesin cetak, mesin pengolah logam dan lain sebagainya [16].

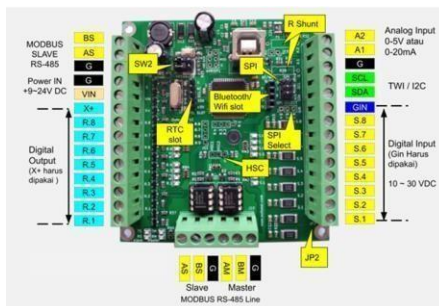
2.5. Webcam



Gambar 5. WebCam

Webcam adalah perangkat keras yang berfungsi sebagai kamera *video digital* dan terhubung ke komputer atau *laptop* melalui *port USB* atau secara nirkabel. Fungsi utama dari *webcam* adalah untuk menangkap dan mengirimkan gambar atau *video* secara langsung melalui jaringan komputer atau internet. *Webcam* biasanya digunakan untuk keperluan seperti konferensi *video*, panggilan *video*, siaran langsung, dan *video streaming*. *Webcam* memiliki resolusi yang berbeda-beda, mulai dari *VGA* hingga *4K*, yang menentukan kualitas gambar atau *video* yang dihasilkan. Beberapa *webcam* juga dilengkapi dengan fitur tambahan seperti *autofocus*, *zoom*, dan *microphone* untuk meningkatkan fungsionalitasnya [17].

2.6. PLC Outseal



Gambar 6. PLC Outseal

PLC Outseal adalah sebuah merek dagang dari Indonesia yang mengembangkan teknologi otomasi, termasuk *PLC (Programmable Logic Controller)* dan *HMI (Human Machine Interface)*. *PLC Outseal* dirancang untuk menjadi teknologi otomasi yang murah, mudah, tangguh, akurat, dan efisien. Salah satu keunggulan dari *PLC Outseal* adalah kompatibilitasnya dengan *Arduino*, yang memungkinkan penggunaan *visual programming* menggunakan diagram tangga dan berbahasa Indonesia untuk pemrogramannya [18]. Berikut adalah beberapa fitur utama dari *PLC Outseal*:

1. *Open Hardware*: *PLC Outseal* dibuat dengan prinsip *open hardware*, yang memungkinkan siapa saja untuk mempelajari, menguji, atau mengembangkan rangkaian elektroniknya.
2. *DIY (Do It Yourself)*: *PLC Outseal* dapat dirakit sendiri di rumah.
3. *Visual Programming*: Pemrograman *PLC Outseal* menggunakan *Outseal Studio*, yang memanfaatkan *visual programming* dengan *ladder diagram*.

2.7. Yolo V5

Metode *YOLO (You Only Look Once)* merupakan teknik pendekatan *computer vision* yang dibuat untuk mendeteksi sebuah objek baik dalam sebuah gambar atau *video* secara langsung dan *real-time*. Sistem pendeteksian pada metode *YOLO* menggunakan pengklasifikasi yang dikonfigurasi ulang, khususnya model yang diterapkan pada gambar menggunakan pendekatan deteksi berbasis area. Dimana pada sebuah pemetaan wilayah pada sebuah gambar yang diberi bobot paling tinggi akan dianggap menjadi sebuah pendeteksian. *YOLO v5*, merupakan versi kelima turunan dari *YOLO*, memiliki pondasi yang serupa dengan *YOLO v4* tetapi menggabungkan banyak penyempurnaan. Akibatnya, *YOLO v5* menunjukkan peningkatan substansial dalam kecepatan operasional, memiliki ukuran yang lebih kecil, dan menunjukkan bobot *file* yang hampir 90% lebih kecil dibandingkan *YOLO v4* [10].

Pada pembuatan alat kali ini akan menggunakan *YOLO v5* dikarenakan akurasi yang tinggi dan algoritma yang mudah dipahami, sehingga lebih unggul dibandingkan metode deteksi lainnya. Hal tersebut dibuktikan melalui penelitian sebelumnya, pada penelitian tersebut disimpulkan bahwa penggunaan metode *YOLO v5* yang digunakan sistem pendekatan untuk mendeteksi cacat pada masker memiliki nilai akurasi tinggi yaitu mencapai 97,1% dan penelitian yang hampir serupa [10].

2.8. Confusion Matrix

Confusion matrix merupakan model klasifikasi yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam pembelajaran mesin. *Confusion matrix* dapat menunjukkan berapa banyak prediksi yang salah dan benar yang dibuat oleh model untuk setiap kelas. *Confusion matrix* dapat digunakan untuk membantu mencari tingkat akurasi dan presisi pada sebuah model [19]. Dalam *confusion matrix* terdapat komponen utama dalam *confusion matrix* untuk klasifikasi,

1. *True Positive (TP)*, yaitu jumlah data positif yang diprediksi benar sebagai positif.
2. *True Negative (TN)*, yaitu jumlah data negatif yang diprediksi benar sebagai negatif.
3. *False Positive (FP)*, yaitu jumlah data negatif yang salah diprediksi sebagai positif (disebut juga *Type I error*).
4. *False Negative (FN)*, yaitu jumlah data positif yang salah diprediksi sebagai negatif (disebut juga *Type II error*).

2.8.1. Akurasi

Akurasi adalah salah satu metrik evaluasi yang digunakan untuk mengukur seberapa baik model klasifikasi dalam memprediksi data dengan benar. Akurasi dihitung dengan membandingkan jumlah prediksi yang benar (untuk kelas positif dan negatif) dengan total keseluruhan [19]. Hal ini digunakan untuk mendapatkan gambaran umum tentang kinerja model klasifikasi. Akurasi menunjukkan proporsi prediksi yang benar dibandingkan dengan keseluruhan prediksi.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2.8.2. Presisi

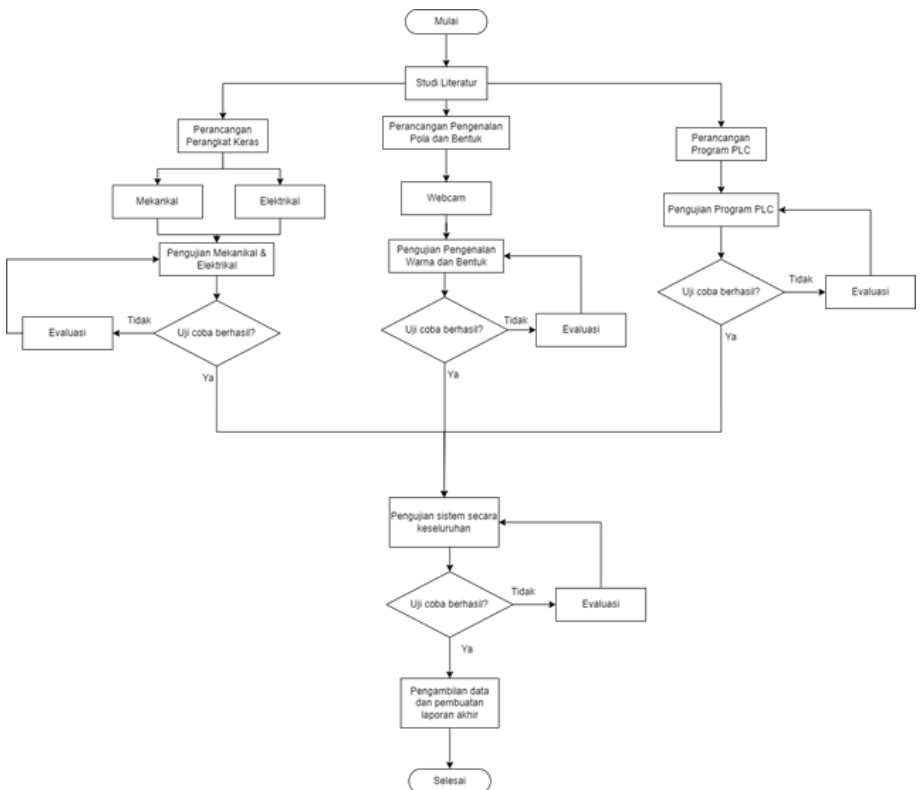
Presisi adalah metrik evaluasi yang menunjukkan seberapa akurat model dalam memprediksi kelas positif. Dengan kata lain, presisi mengukur proporsi prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model [19]. Hal ini dapat membantu untuk menghindari prediksi positif yang salah.

$$Presisi = \frac{TP}{TP + FP} \quad (2)$$

Bab 3. Metodologi Penelitian

3.1. Perancangan

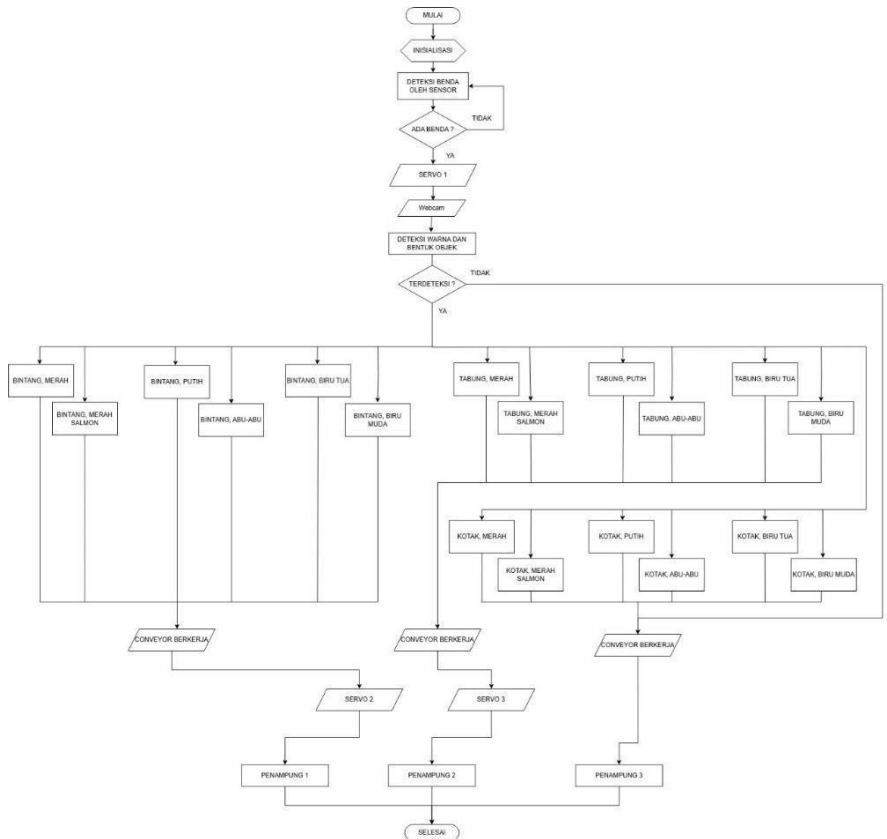
Perancangan sistem penyortiran sistem sortir barang berdasarkan warna dan bentuk menggunakan *webcam* berbasis *PLC* dimulai dari perancangan perangkat keras yaitu proses menghubungkan beberapa komponen mulai dari mekanikal hingga elektrikal, perancangan program *PLC* terdiri dari pemrograman perangkat kontrol *outseal* menggunakan *Ladder Diagram*, pengujian sistem yaitu menguji dari sistem sesuai tabel pengujian, pengambilan data yaitu proses untuk mengambil beberapa sampel dari pengujian sistem, dan pembuatan laporan akhir.



Gambar 7. Diagram Alur Pengerjaan

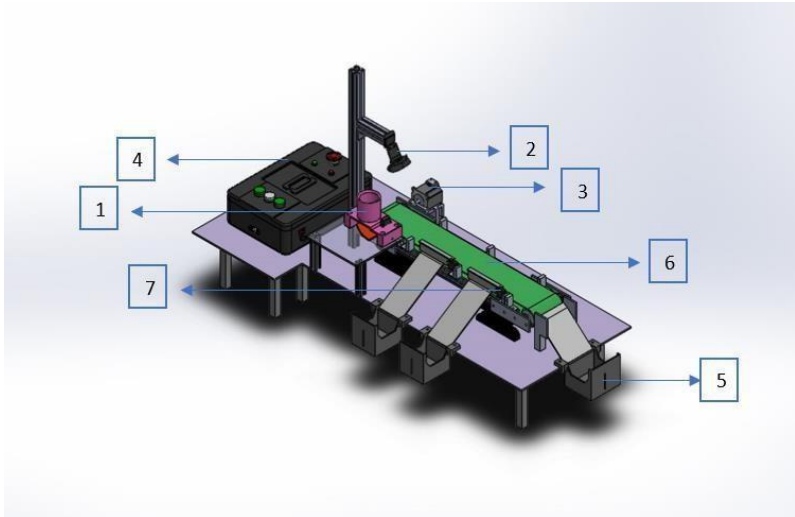
3.1.1. Perancangan Sistem Kerja

Berikut ini adalah *flowchart* dari sistem alat dan langkah kerja dari alat, adapun *flowchart* alat ini dapat dilihat pada gambar dibawah ini.



Gambar 8. Diagram Alur Sistem Kerja

3.1.2. Perancangan Sistem Mekanik

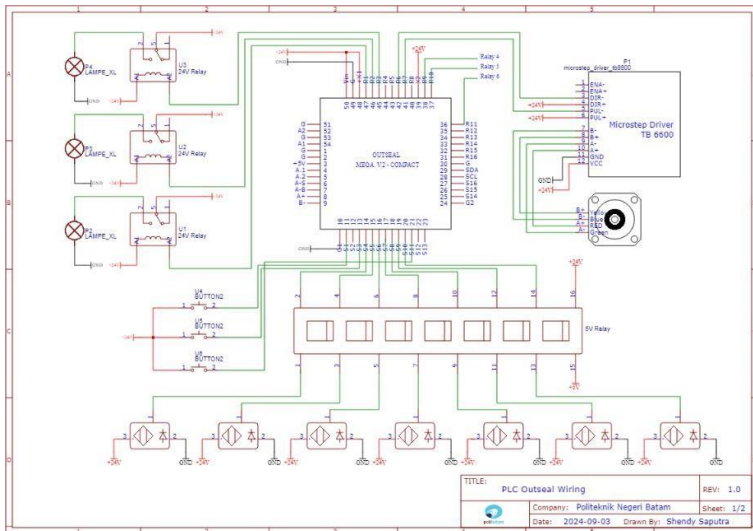


Gambar 9. Design Alat

Alat ini dibangun dengan kontrol utama *PLC Outseal* yang tertanam dalam kotak yang disesuaikan. *Webcam* berada di depan jalur awal *conveyor*. Ketika sebuah objek melintas melalui *conveyor*, *webcam* akan memproses bentuk dan warna objek tersebut. Setelah objek diklasifikasikan, motor *servo* mengarahkan objek ke tempat penampungan yang telah dipersiapkan dengan menutup jalur yang dilewatinya. Berikut keterangan dari beberapa komponen pada alat diatas:

1. Tabung Silinder
2. Kamera/*Webcam*
3. Motor *Stepper*
4. *Box Panel*
5. *Storage Box*
6. *Conveyor*
7. Motor *Servo*

3.1.3. Perancangan Sistem Elektrikal

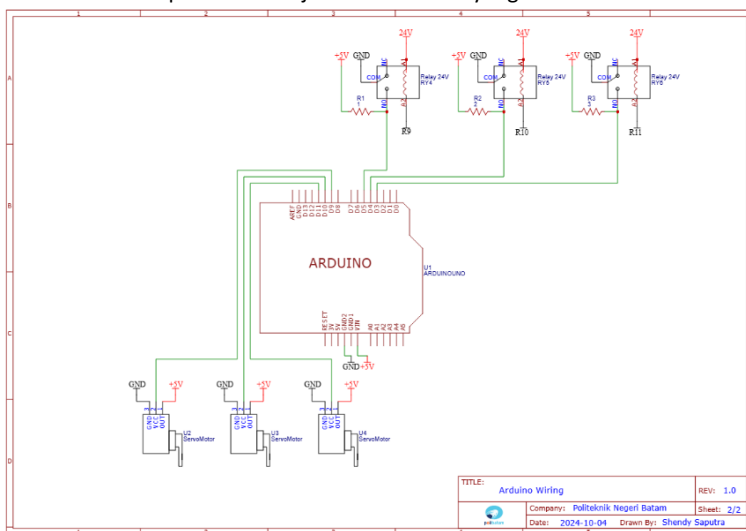


Gambar 10. Schematic Wiring PLC Outseal

Pada gambar diatas merupakan rancangan rangkaian elektrikal sistem yang terhubung pada *PLC Outseal* sebagai sistem kendali utama pada alat ini. Secara keseluruhan, diagram diatas menunjukkan bagaimana *PLC* dapat digunakan untuk mengontrol *relay*, lampu, tombol, dan motor *stepper* melalui berbagai komponen dan koneksi yang ada. Setiap komponen yang digunakan diatur sedemikian rupa sehingga sistem dapat bekerja secara otomatis sesuai dengan *ladder diagram* yang diprogram di dalam *PLC*.

Pada *wiring diagram PLC Outseal*, terdapat *pin input* dan *pin output* yang telah terhubung pada beberapa komponen yang berfungsi untuk mengontrolnya. *PLC Outseal* diaktifkan dengan memberikan tegangan 24V pada *pin Vin*, sementara untuk mengaktifkan *input*, *pin G1* dihubungkan ke *ground*. *Output PLC* akan aktif jika tegangan 24V diberikan pada *pin X1* dan *X2*. Tombol darurat (*emergency*) terhubung ke *pin input S1*, tombol start terhubung ke *S11*, dan tombol stop terhubung ke *S3*. Selain itu, *pin input S4* hingga *S10* dihubungkan ke *output relay 5V*. Pada bagian *output PLC*, beberapa *pin* digunakan untuk mengendalikan *relay* dan motor *stepper*, yaitu *R1* hingga *R4*, *R6*, *R7*, serta *R9* hingga *R11*. *Pin R1* terhubung ke *A2* pada *relay U1*, *R2* ke *A2* pada *relay U2*, *R3* ke *A2* pada *relay U3*. *Pin R6* dan *R7* digunakan untuk mengontrol *driver motor stepper* dengan menghubungkannya masing-masing ke *Dir-* dan *Pul-*. Sementara itu, *R9*, *R10*, dan *R11* terhubung ke *A2* pada *relay 4*, *relay 5*, dan *relay 6*. Dan *Output* dari *5V relay*

akan terhubung pada sensor *proximity*. Sebanyak tujuh sensor yang digunakan untuk mendeteksi objek dimana sensor-sensor yang digunakan dalam sistem ini memiliki fungsi masing-masing untuk mendeteksi keberadaan dan posisi benda di berbagai tahapan. Sensor 1 berfungsi mendeteksi keberadaan benda di dalam dan dibawah silinder. Sensor 2 memonitor keberadaan benda pada konveyor tepatnya dibawah kamera, saat benda melewati area pemilahan pertama, Sensor 3 mendeteksinya, sementara Sensor 4 mengonfirmasi bahwa benda telah terpilah dan berada di tempat pemilahan pertama. Selanjutnya, Sensor 5 mendeteksi benda yang lewat di area penampungan pertama, dan Sensor 6 memastikan benda tersebut telah terpilah dan berada di tempat pemilahan kedua. Terakhir, Sensor 7 memverifikasi bahwa benda sudah terpilah dan berada di lokasi penampungan kedua. Dengan konfigurasi ini, sensor-sensor tersebut mendukung proses otomatisasi pemilahan objek sesuai urutan yang telah ditentukan.



Gambar 11. Schematic Wiring Arduino Uno

Pada gambar diatas merupakan rancangan rangkaian elektrikl sistem yang terhubung pada *Arduino UNO*. Digunakan mikrokontroler ini karena untuk memudahkan penggerakkan motor *servo* dalam pengaturan derajat pada *servo* melalui program bahasa *C* dengan *Aduino IDE*. *Arduino* diaktifkan oleh teggangan 5V, pada *pin D2* hingga *D5* digunakan untuk mengontrol tiga *relay*. Setiap *pin output Arduino* ini akan mengaktifkan *relay* saat dalam kondisi *high*. Pada teggangan 24V digunakan untuk memberikan teggangan pada *pin A1* dari *relay* dan *pin A2* akan diaktifkan oleh *PLC Outseal*.

3.1.4. Perancangan Sistem Pendeteksi Bentuk dan Warna

Berikut ini merupakan alur kerja atau *flowchart* pembuatan sistem pendeteksi warna menggunakan metode *YOLO v5*.



Gambar 12. Diagram Alur Sistem Pendeteksi Bentuk dan Warna

Proses deteksi objek pada diagram alir di atas dimulai dari tahap awal yang disebut Mulai. Setelah itu, pengguna melakukan memilih Bobot, yaitu memilih model atau bobot deteksi objek yang sesuai. Bobot ini berfungsi sebagai dasar dalam algoritma yang akan digunakan untuk mengenali pola objek. Selanjutnya, dilakuakn deteksi objek pada gambar atau deteksi objek secara *real-time*. Proses berlanjut ke tahap menjalankan sistem, di mana sistem memulai deteksi objek sesuai dengan pilihan yang telah dibuat. Hasil dari proses ini akan ditampilkan pada langkah hasil deteksi.

3.2. Alat dan Bahan

Tabel 2. Estimasi Biaya



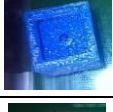

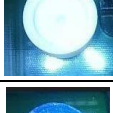
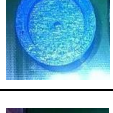

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Ket.
1.	Gergaji	30.000	1 pc	30.000	
2.	Gerinda	600.000	1 pc	600.000	
3.	Obeng <i>Plus</i>	10.000	2 pcs	20.000	
4.	Obeng <i>Minus</i>	10.000	2 pcs	20.000	
5.	Palu	50.000	1 pc	50.000	
6.	PC/Laptop	-	1 pc	-	
7.	Akrilik (3mm) 1x2 m	500.000	1x2 meter	500.000	
8.	<i>Aluminium Profile</i> 20x20 mm (1.5m)	130.000	3 meter	260.000	
9.	Konveyor	500.000	1	500.000	
10.	Sekrup	10.000	1 box	10.000	
11.	Baut + Mur Baut	20.000	1 box	20.000	
12.	<i>Ring baut</i>	5.000	1 box	5.000	
13.	<i>Adapter 24v</i>	100.000	1 pc	100.000	
14.	<i>Stepper Motor NEMA23</i>	230.000	1 pc	230.000	
15.	<i>Microstep Driver TB6600</i>	70.000	1 pc	70.000	
16.	<i>Micro DC Servo Motor</i>	50.000	3 pcs	150.000	
17.	<i>Sensor Optik</i>	250.000	6 pcs	1.500.000	












18.	<i>Sensor Proximity</i>	100.000	1 pc	100.000	
19.	Modul Relay 24 V	30.000	7 pcs	180.000	
20.	Relay	60.000	6 pcs	360.000	
21.	<i>Webcam</i>	150.000	1 pc	150.000	
22.	<i>PLC Outseal V.2</i>	500.000	1 pc	500.000	
23.	Kabel (1mm)	250.000	1 gulung	250.000	
24.	Lem	5.000	1 pc	5.000	
25.	<i>Push Button</i> (Merah)	50.000	1 pc	50.000	
26.	<i>Push Button</i> (Hijau)	50.000	1 pc	50.000	
27.	<i>Emergency Button</i>	50.000	1 pc	50.000	
28.	<i>Pilot Light</i> (Merah, Kuning, Hijau)	25.000	3 pcs	75.000	
29.	Besi <i>Hollow</i> 2x2 cm 1.2mm (6m)	60.000	3 btng	180.000	
30.	Filamen	60.000	3 gulung	840.000	
31.	<i>Ring Light</i>	150.000	1 pc	100.000	
	TOTAL			6.855.000	



3.3. Pengujian

Alat ini didesain dengan sebuah *webcam* yang terletak di atas konveyor yang berdekatan dengan tabung penampung yang mampu menampung sekitar 3-4 benda. Dan diatas *webcam* ditambahkan sebuah lampu yang digunakan sebagai pencahayaan untuk mendukung proses deteksi. Alat ini menggunakan motor *servo* sebagai pemisah objek yang diklasifikasikan. Dalam pengujian, alat ini menggunakan 27 objek yang berbeda. Berikut adalah tabel klasifikasi benda yang akan diuji.

Tabel 3. Klasifikasi Benda yang Diuji

No	Nama Objek	Warna	Bentuk	Gambar
1	kubus_merah_only	Merah	Kubus	
2	kubus_putih	Putih	Kubus	
3	kubus_biru_tua	Biru	Kubus	
4	tabung_merah_only	Merah	Tabung	
5	tabung_putih	Putih	Tabung	
6	tabung_biru_tua	Biru	Tabung	
7	bintang_merah_only	Merah	Bintang	

8	bintang_putih	Putih	Bintang	
9	bintang_biru_tua	Biru	Bintang	
10	kubus_merah_salmon	Merah Salmon	Kubus	
11	kubus_abu_abu	Abu-abu	Kubus	
12	kubus_biru_muda	Biru muda	Kubus	
13	tabung_merah_salmon	Merah Salmon	Tabung	
14	tabung_abu_abu	Abu-abu	Tabung	
15	tabung_biru_muda	Biru muda	Tabung	
16	bintang_merah_salmon	Merah Salmon	Bintang	
17	bintang_abu_abu	Abu-abu	Bintang	
18	bintang_biru_muda	Biru muda	Bintang	

19	objek_1	Abu-abu & pink	Bungkus permen	
20	objek_2	Hijau Opal	Segitiga	
21	objek_3	Hijau Opal	Limas	
22	objek_4	Hijau	Rautan	
23	objek_5	Merah	Lilin	
24	objek_6	Hijau Opal	Tabung tak sempurna	
25	objek_7	Hijau Opal	Segi enam tak sempurna	
26	objek_8	Transparan	Oval	
27	objek_9	Kuning	Dua lingkaran	

3.3.1. Pengujian Deteksi Adanya Benda

Pada pengujian ini, proses dilakukan untuk memastikan bahwa objek yang ada dalam sistem dapat dideteksi secara akurat oleh sensor. Langkah ini sangat penting agar sistem dapat mengenali keberadaan objek tersebut dan memastikan bahwa sensor dapat mengidentifikasi objek dengan baik. Ketika objek terdeteksi, sensor akan mengirimkan sinyal yang memicu mekanisme dorongan untuk memindahkan objek tersebut ke atas *conveyor*. Selanjutnya, objek yang sudah berada di *conveyor* akan dipindai oleh *webcam*. *Webcam* berperan sebagai perangkat visual yang menangkap gambar atau *video* dari objek, memungkinkan sistem untuk melakukan analisis visual yang lebih mendalam. Data visual yang ditangkap oleh *webcam* kemudian diproses untuk mengidentifikasi karakteristik objek, seperti bentuk, warna, atau ukuran.

Berdasarkan informasi ini, sistem akan mengklasifikasikan objek sesuai dengan kategori atau kriteria yang telah ditentukan. Untuk objek berbentuk bintang akan dipisahkan ke penampung 1 dengan *servo* 2. Untuk objek berbentuk tabung akan dipisahkan ke penampung 2 dengan *servo* 3. Sedangkan untuk objek berbentuk kubus akan dipisahkan ke penampung 3. Pengujian ini tidak hanya memastikan bahwa sensor berfungsi dengan baik dalam mendeteksi dan menggerakkan objek ke *conveyor*, tetapi juga bahwa *webcam* dapat mendeteksi objek secara akurat untuk klasifikasi yang lebih lanjut.

3.3.2. Pengujian Deteksi Bentuk Dan Warna

Pada pengujian ini, prosedur dilakukan untuk memastikan bahwa sistem dapat mendeteksi bentuk dan warna objek dengan tingkat akurasi yang tinggi. Proses ini bertujuan untuk mengidentifikasi dan memverifikasi bahwa sistem deteksi dapat mengenali bentuk serta warna objek secara konsisten dan sesuai dengan kriteria yang telah ditentukan. Deteksi objek dilakukan di metode *detect_objects*. Pada bagian ini, *YOLO* menerima *input* berupa gambar dan menghasilkan prediksi berupa *bounding box*, *confidence score*, dan *class label*.

```
def detect_objects(self, frame, classes, net, x_scale, y_scale):
    blob = cv2.dnn.blobFromImage(frame, scalefactor=1/255, size=[INPUT_SIZE, INPUT_SIZE],
                                mean=[0, 0, 0], swapRB=False, crop=False)
    net.setInput(blob)
    detections_raw = net.forward()[0]
```

Gambar 13. Program *Python* Pendeteksi Objek

Tahap awal dari pengujian ini melibatkan pemantauan respons sistem saat menghadapi berbagai macam bentuk dan warna objek. Sensor serta algoritma pengenalan gambar yang terlibat harus mampu membedakan berbagai karakteristik objek ini tanpa kesalahan, sehingga memungkinkan deteksi yang

presisi. Ketika bentuk dan warna objek berhasil dideteksi, hasil deteksi tersebut akan menjadi dasar untuk proses selanjutnya, yaitu pengklasifikasian objek.

```
self.classes = ['bintang_abu_abu', 'bintang_biru_muda', 'bintang_biru_tua', 'bintang_merah_only',  
               'bintang_merah_salmon', 'bintang_putih', 'kubus_abu_abu', 'kubus_biru_muda',  
               'kubus_biru_tua', 'kubus_merah_only', 'kubus_merah_salmon', 'kubus_putih',  
               'tabung_abu_abu', 'tabung_biru_muda', 'tabung_biru_tua', 'tabung_merah_only',  
               'tabung_merah_salmon', 'tabung_putih']
```

Gambar 14. Program *Python* Pendeteksi Bentuk dan Warna

Deteksi warna ditentukan berdasarkan nama warna dalam daftar kelas (misalnya: abu_abu, biru_muda, dan lain lain). Deteksi bentuk ditentukan berdasarkan nama bentuk dalam daftar kelas (misalnya: bintang, kubus, tabung). Pada tahap pengklasifikasian, sistem akan menggunakan data yang diperoleh dari deteksi awal untuk menempatkan objek ke dalam kategori yang sesuai berdasarkan bentuk dan warnanya. Pengujian ini tidak hanya bertujuan untuk memvalidasi kemampuan sensor dan algoritma dalam mendeteksi bentuk serta warna, tetapi juga untuk mengonfirmasi bahwa setiap hasil deteksi dapat dilanjutkan ke tahap klasifikasi dengan data yang tepat.

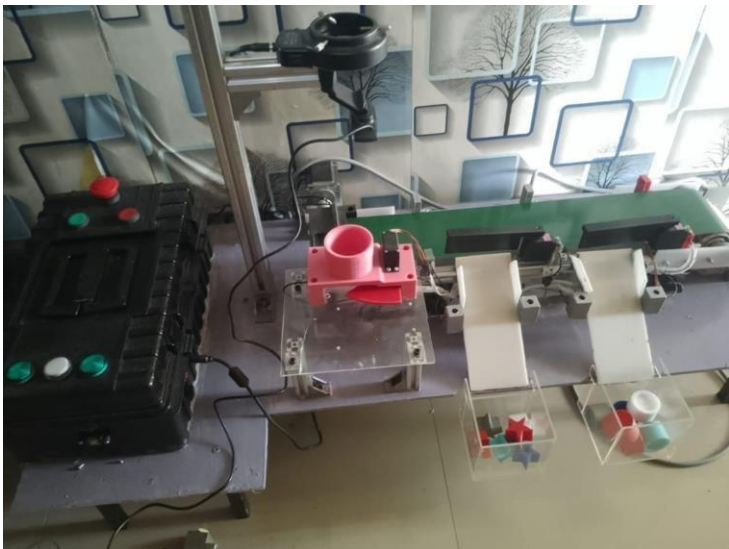
Bab 4. Hasil dan Pembahasan

4.1. Hasil yang Telah Dicapai

4.1.1. Alat Pemilah Benda

Pada tahap akhir, terbentuklah alat mekanis berupa *conveyor* pemisah untuk mengelompokkan benda-benda berdasarkan klasifikasi tertentu. Proses klasifikasi ini dilakukan secara otomatis dengan menggunakan sistem yang dilengkapi teknologi pengenalan objek. Di atas *conveyor*, dipasang sebuah kamera yang berfungsi mendeteksi dan mengidentifikasi karakteristik setiap benda yang melewatinya.

Kamera tersebut akan menangkap citra benda dan mengirimkannya ke sistem untuk dianalisis. Berdasarkan hasil analisis, benda yang telah terklasifikasi akan diteruskan ke tahap penyortiran. Di tahap ini, motor *servo* akan berfungsi mengarahkan benda ke jalur yang sesuai dengan klasifikasi yang telah ditentukan, sehingga benda tersebut jatuh ke dalam wadah penampung yang sesuai dengan kategorinya.

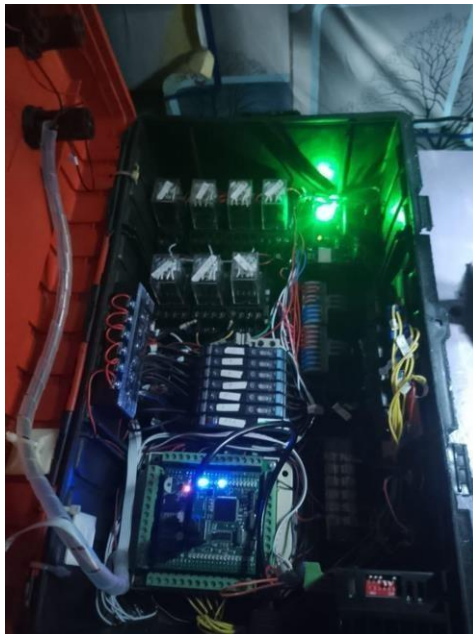


Gambar 15. Hasil dari Perancangan Mekanikal

4.1.2. Kelistrikan Pada Alat Pemilah Benda

Berdasarkan perancangan elektrikal yang telah dilakukan sebelumnya, terbentuklah sistem kontrol untuk alat pemilah benda ini. Setiap komponen berhasil dirangkai dan dihubungkan dalam satu sistem terpadu yang memungkinkan seluruh proses pemilahan berjalan secara otomatis dan terkontrol dengan baik. Sistem kendali ini diatur menggunakan *PLC Outseal*, yang berfungsi sebagai pusat pengaturan bagi seluruh komponen yang terpasang.

Pada tahap akhir, semua komponen dapat beroperasi secara sinkron melalui sistem kontrol yang telah diatur dalam *PLC* tersebut. Setiap instruksi yang telah diprogram di dalam *PLC Outseal* memungkinkan motor dan sensor pada alat pemilah benda ini bekerja sesuai dengan logika pemilahan yang dirancang. Dan *servo* dapat berfungsi sebagai pemisah benda yang telah diklasifikasikan sebelumnya dengan sesuai yang diperintahkan oleh *Arduino* yang digunakan. Dengan demikian, sistem dapat melakukan deteksi, klasifikasi, serta pemindahan benda ke lokasi penampung yang sesuai secara otomatis dan presisi.



Gambar 16. Hasil dari Perancangan Elektrikal

4.1.3. Hasil Pengujian Deteksi Adanya Benda

Pengujian ini dilakukan untuk mendeteksi keberadaan benda pada tempat peletakan awal. Ketika sensor mendeteksi adanya benda, *servo 1* akan mendorong benda tersebut ke *conveyor* di bawah kamera agar dapat dideteksi oleh kamera dan diklasifikasikan lebih lanjut. Pada *servo 2* objek akan terdorong pada tempat penampung pertama dan *servo 3* objek akan terdorong ke penampung 2. Berikut adalah hasil dari pengujian yang telah dilakukan.

Tabel 4. Hasil Deteksi Adanya Benda

No	Nama Objek	Warna	Bentuk	Terdeteksi oleh Sensor (Y/N)	Ket
1	kubus_merah_only	Merah	Kubus	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
2	kubus_putih	Putih	Kubus	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
3	kubus_biru_tua	Biru	Kubus	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
4	tabung_merah_only	Merah	Tabung	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
					<i>Servo 3</i> , dapat memisahkan benda ke penampung 2
5	tabung_putih	Putih	Tabung	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
					<i>Servo 3</i> , dapat memisahkan benda ke penampung 2

6	tabung_biru_tua	Biru	Tabung	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 3, dapat memisahkan benda ke penampung 2
7	bintang_merah_only	Merah	Bintang	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 2, dapat memisahkan benda ke penampung 1
8	bintang_putih	Putih	Bintang	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 2, dapat memisahkan benda ke penampung 1
9	bintang_biru_tua	Biru	Bintang	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 2, dapat memisahkan benda ke penampung 1
10	kubus_merah_salmon	Merah Salmon	Kubus	Y	Servo 1, dapat mendorong objek tepat dibawah kamera

11	kubus_abu_abu	Abu-abu	Kubus	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
12	kubus_biru_muda	Biru muda	Kubus	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
13	tabung_merah_salmon	Merah Salmon	Tabung	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
					<i>Servo 3</i> , dapat memisahkan benda ke penampung 2
14	tabung_abu_abu	Abu-abu	Tabung	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
					<i>Servo 3</i> , dapat memisahkan benda ke penampung 2
15	tabung_biru_muda	Biru muda	Tabung	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera
					<i>Servo 3</i> , dapat memisahkan benda ke penampung 2
16	bintang_merah_salmon	Merah Salmon	Bintang	Y	<i>Servo 1</i> , dapat mendorong objek tepat dibawah kamera

					Servo 2, dapat memisahkan benda ke penampung 1
17	bintang_abu_abu	Abu-abu	Bintang	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 2, dapat memisahkan benda ke penampung 1
18	bintang_biru_muda	Biru muda	Bintang	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
					Servo 2, dapat memisahkan benda ke penampung 1
19	objek_1	Abu-abu & pink	Bungkus permen	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
20	objek_2	Hijau Opal	Segitiga	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
21	objek_3	Hijau Opal	Limas	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
22	objek_4	Hijau	Rautan	Y	Servo 1, dapat mendorong objek tepat dibawah kamera

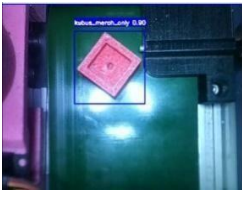
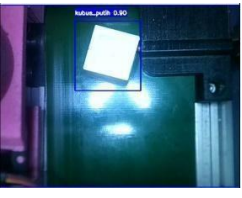
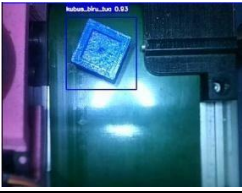
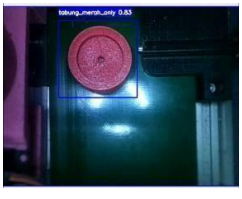
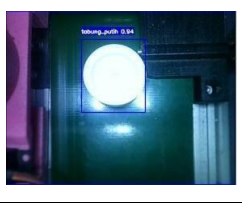
23	objek_5	Merah	Lilin	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
24	objek_6	Hijau Opal	Tabung tak sempurna	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
25	objek_7	Hijau Opal	Segi enam tak sempurna	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
26	objek_8	Transparan	Oval	Y	Servo 1, dapat mendorong objek tepat dibawah kamera
27	objek_9	Kuning	Dua lingkaran	Y	Servo 1, dapat mendorong objek tepat dibawah kamera

4.1.4. Pengujian Deteksi Bentuk Dan Warna

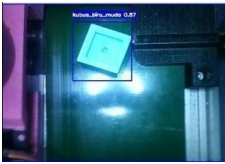





Berikut adalah hasil pengujian untuk deteksi objek menggunakan kamera yang telah dioptimalkan melalui berbagai tahap pengambilan data. Pengujian dilakukan pada tiga kondisi cahaya yang berbeda yaitu, cahaya pada saat siang hari, sore hari, dan menggunakan tambahan *ringlight* di atas *webcam* pada lokasi pengujian yang sama. Untuk memastikan kamera dapat mendeteksi objek dengan akurat, telah dikumpulkan data pelatihan sebanyak 200 kali melalui aplikasi *Roboflow*. Aplikasi ini memudahkan proses pengelolaan dan pengolahan dataset gambar, termasuk tahap pelabelan objek menggunakan fitur yang disebut *Dataset Annotation*.



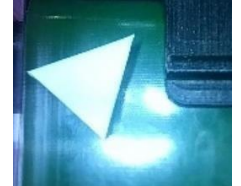



Dengan fitur *Dataset Annotation*, pengguna dapat mengunggah sejumlah gambar sekaligus dan melakukan anotasi pada setiap gambar. Proses anotasi ini meliputi penandaan objek dan pemberian label pada setiap jenis objek. Fitur ini mendukung peningkatan akurasi sistem dalam mengenali objek, serta memungkinkan pelatihan model deteksi yang lebih optimal. Berikut adalah hasil pengujian untuk deteksi bentuk dan warna pada objek.



Tabel 5. Hasil Deteksi Bentuk dan Warna dengan *Ringlight*

No	Nama Objek	Hasil Deteksi Kamera	Sesuai (Y/N)
1	kubus_merah_only		Y
2	kubus_putih		Y
3	kubus_biru_tua		Y
4	tabung_merah_only		Y
5	tabung_putih		Y

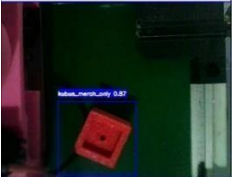

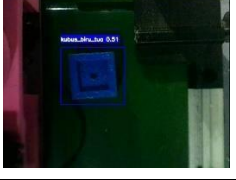


6	tabung_biru_tua		Y
7	bintang_merah_only		Y
8	bintang_putih		Y
9	bintang_biru_tua		Y
10	kubus_merah_salmon		Y
11	kubus_abu_abu		Y


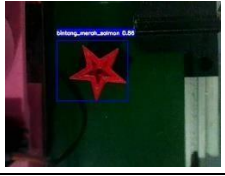
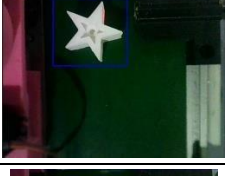
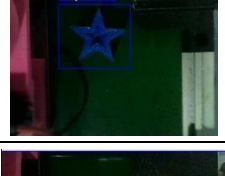


12	kubus_biru_muda		Y
13	tabung_merah_salmon		Y
14	tabung_abu_abu		Y
15	tabung_biru_muda		Y
16	bintang_merah_salmon		Y
17	bintang_abu_abu		Y




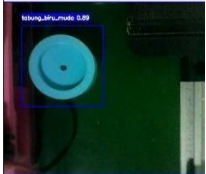


18	bintang_biru_muda		Y
19	objek_1		Y
20	objek_2		Y
21	objek_3		Y
22	objek_4		Y
23	objek_5		Y

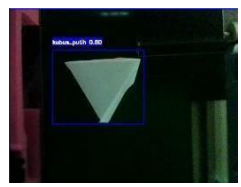
24	objek_6		Y
25	objek_7		Y
26	objek_8		Y
27	objek_9		Y





Tabel 6. Hasil Deteksi Bentuk dan Warna pada Siang Hari

No	Nama Objek	Hasil Deteksi Kamera	Sesuai (Y/N)
1	kubus_merah_only		Y
2	kubus_putih		Y
3	kubus_biru_tua		Y
4	tabung_merah_only		Y
5	tabung_putih		Y





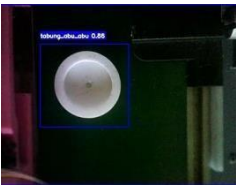
6	tabung_biru_tua		Y
7	bintang_merah_only		N
8	bintang_putih		Y
9	bintang_biru_tua		Y
10	kubus_merah_salmon		Y
11	kubus_abu_abu		N

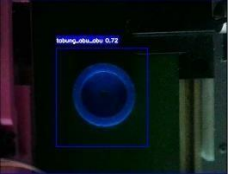



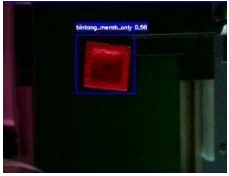

12	kubus_biru_muda		Y
13	tabung_merah_salmon		Y
14	tabung_abu_abu		Y
15	tabung_biru_muda		Y
16	bintang_merah_salmon		Y
17	bintang_abu_abu		Y

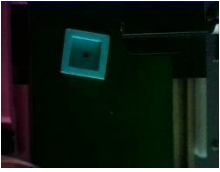
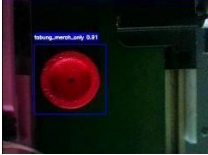

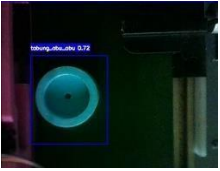


18	bintang_biru_muda		Y
19	objek_1		Y
20	objek_2		N
21	objek_3		N
22	objek_4		Y
23	objek_5		Y




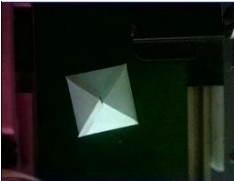


24	objek_6		Y
25	objek_7		Y
26	objek_8		Y
27	objek_9		Y



Tabel 7. Hasil Deteksi Bentuk dan Warna pada Sore Hari

No	Nama Objek	Hasil Deteksi Kamera	Sesuai (Y/N)
1	kubus_merah_only		N
2	kubus_putih		N
3	kubus_biru_tua		N
4	tabung_merah_only		Y
5	tabung_putih		N

6	tabung_biru_tua		N
7	bintang_merah_only		N
8	bintang_putih		N
9	bintang_biru_tua		N
10	kubus_merah_salmon		N
11	kubus_abu_abu		N

12	kubus_biru_muda		N
13	tabung_merah_salmon		N
14	tabung_abu_abu		N
15	tabung_biru_muda		N
16	bintang_merah_salmon		Y
17	bintang_abu_abu		Y



18	bintang_biru_muda		Y
19	objek_1		Y
20	objek_2		Y
21	objek_3		Y
22	objek_4		Y
23	objek_5		Y


24	objek_6		Y
25	objek_7		Y
26	objek_8		Y
27	objek_9		Y

4.1.5. Pengujian *Sequence* (Proses) Pada Alat Pemilah Benda

Pengujian ini dilakukan untuk memastikan bahwa proses pemilahan benda berjalan dengan baik dan runtut, mulai dari awal hingga barang masuk ke tempat penampung sesuai dengan klasifikasinya yang dilakukan sebanyak 25 kali percobaan dengan urutan objek yang acak. Dalam pengujian ini didapatkan hasil akurasi dan presisi 100% yang mengindikasikan bahwa proses pemilahan benda berjalan dengan sangat baik dan runtut. Berikut adalah tabel hasil pengujiannya.

Tabel 8. Hasil Pengujian *Sequence* (Proses) Pada Alat Pemilah Benda

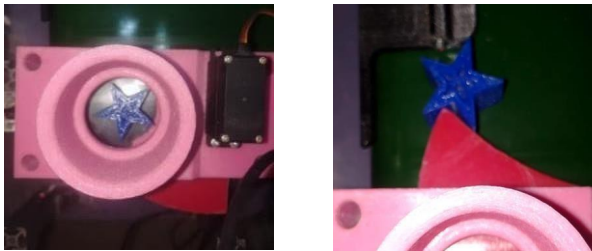
<i>Sequence</i>	Jumlah Percobaan	Percobaan Berhasil	Percobaan Gagal	Tingkat Keberhasilan	Ket. Gambar
Sistem kontrol bekerja sesuai fungsinya masing-masing (inisialisasi alat pemilah benda)	25	25	0	100%	
Pendeteksian adanya objek masuk pada tempat peletakkan objek	25	25	0	100%	
Motor <i>servo</i> (1) mendorong objek agar berada tepat di bawah <i>webcam</i>	25	25	0	100%	
<i>Webcam</i> mendeteksi objek yang masuk dan mengklasifikasikannya sesuai dengan ketentuan yang sudah ditetapkan	25	25	0	100%	
Motor <i>stepper</i> menggerakkan konveyor untuk membawa objek	25	25	0	100%	
Sensor klasifikasi objek mendeteksi objek sesuai dengan klasifikasi yang sudah ditentukan & motor <i>servo</i> (2 & 3) mendorong objek agar masuk ke penampungnya masing-masing. (Khusus objek kubus akan dibawa oleh konveyor sampai ke penampung)	25	25	0	100%	

Sensor optik membaca adanya objek yang masuk ke penampung (1 & 2)	25	25	0	100%	
---	----	----	---	------	---

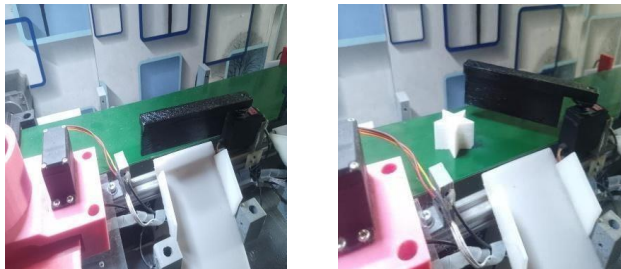
4.2. Pembahasan

4.2.1. Pembahasan Hasil Pengujian Deteksi Adanya Benda

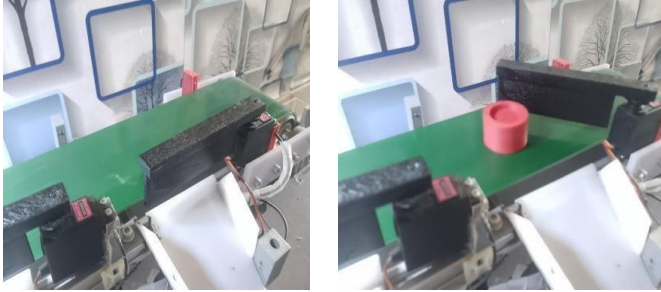
Pada pengujian sistem deteksi objek yang ditempatkan di area peletakan awal, seluruh objek berhasil terdeteksi dengan baik oleh sensor yang digunakan. Keberhasilan deteksi ini ditunjukkan dengan adanya indikator yang terhubung pada sensor, yang akan mati apabila ada objek yang berada di tempat peletakan awal. Hal ini memastikan bahwa sensor berfungsi dengan baik dalam mengenali keberadaan objek pada posisi tersebut. Selain itu, keberhasilan pengujian ini juga terlihat ketika *servo* mampu mendorong objek yang telah terdeteksi oleh sensor menuju titik awal jalur konveyor dan pada penampungnya. Pada titik awal konveyor ini, terdapat kamera yang siap untuk melaksanakan proses selanjutnya.



Gambar 17. Hasil dari Pengujian Deteksi Benda dan Pergerakan *Servo* 1



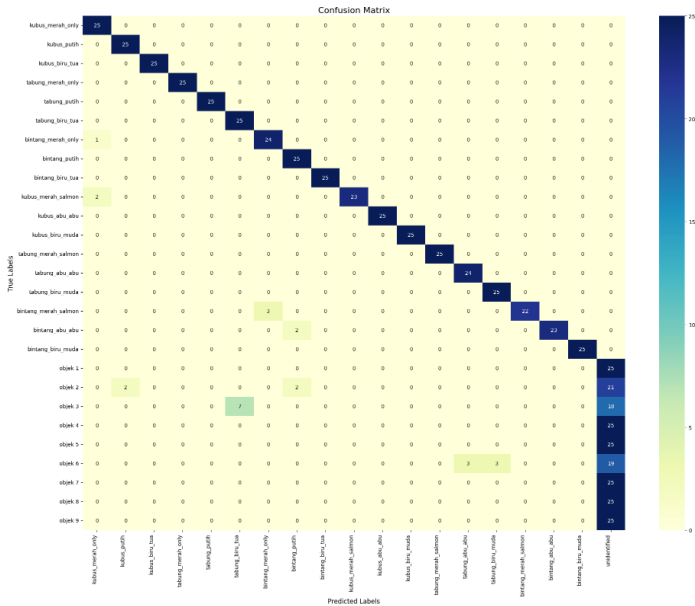
Gambar 18. Hasil dari Pengujian Deteksi Benda dan Pergerakan *Servo* 2



Gambar 19. Hasil dari Pengujian Deteksi Benda dan Pergerakan Servo 3

4.2.2. Pembahasan Hasil Pengujian Deteksi Bentuk dan Warna

Pada tahap pengujian deteksi objek untuk mendeteksi bentuk dan warna yang dilakukan sebanyak 25 kali percobaan pada setiap jenis objek yang tersedia. Pengujian dilakukan pada tiga kondisi cahaya yang berbeda yaitu, cahaya pada saat siang hari, sore hari, dan menggunakan tambahan *ringlight* di atas *webcam* pada lokasi pengujian yang sama. Dalam proses ini, digunakan model klasifikasi *confusion matrix*. *Confusion matrix* memungkinkan evaluasi kinerja sistem secara lebih rinci, dengan memberikan informasi mengenai jumlah deteksi yang benar (*true positives*), kesalahan deteksi (*false positives dan false negatives*), serta kesalahan penolakan (*true negatives*). Data yang dihasilkan dari *confusion matrix* ini kemudian dianalisis untuk mengetahui akurasi dan presisinya.



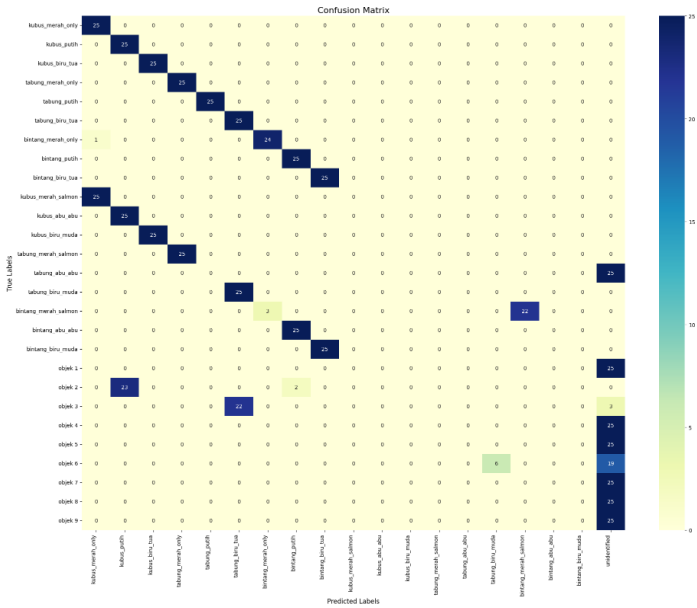
Gambar 20. Model Klasifikasi *Confusion Matrix* pada Hasil Pengujian Deteksi Bentuk dan Warna menggunakan *Ringlight*

Berdasarkan hasil klasifikasi dari model *confusion matrix* pada hasil pengujian deteksi bentuk dan warna menggunakan *ringlight* diatas dapat disimpulkan pada tabel di bawah ini.

Tabel 9. Hasil Klasifikasi Deteksi Bentuk dan Warna menggunakan *Ringlight*

No	Nama Objek	Keterangan
1	kubus_merah_only	Tidak ada kesalahan sama sekali
2	kubus_putih	Tidak ada kesalahan sama sekali
3	kubus_biru_tua	Tidak ada kesalahan sama sekali
4	tabung_merah_only	Tidak ada kesalahan sama sekali
5	tabung_putih	Tidak ada kesalahan sama sekali
6	tabung_biru_tua	Tidak ada kesalahan sama sekali
7	bintang_merah_only	Kesalahan deteksi sebagai objek kubus satu kali
8	bintang_putih	Tidak ada kesalahan sama sekali

9	bintang_biru_tua	Tidak ada kesalahan sama sekali
10	kubus_merah_salmon	Kesalahan deteksi berwarna merah dua kali
11	kubus_abu_abu	Tidak ada kesalahan sama sekali
12	kubus_biru_muda	Tidak ada kesalahan sama sekali
13	tabung_merah_salmon	Tidak ada kesalahan sama sekali
14	tabung_abu_abu	Kesalahan deteksi sebagai objek berwarna putih satu kali
15	tabung_biru_muda	Tidak ada kesalahan sama sekali
16	bintang_merah_salmon	Kesalahan deteksi sebagai objek berwarna merah tiga kali
17	bintang_abu_abu	Kesalahan deteksi sebagai objek berwarna putih dua kali
18	bintang_biru_muda	Tidak ada kesalahan sama sekali
19	objek_1	Tidak ada kesalahan sama sekali
20	objek_2	Kesalahan deteksi sebagai bintang_putih sebanyak 2 kali dan kubus_putih
21	objek_3	Kesalahan deteksi sebagai tabung_biru_tua sebanyak 7 kali
22	objek_4	Tidak ada kesalahan sama sekali
23	objek_5	Tidak ada kesalahan sama sekali
24	objek_6	Kesalahan deteksi sebagai tabung_biru_muda sebanyak 3 kali dan tabung_abu_abu sebanyak 3 kali
25	objek_7	Tidak ada kesalahan sama sekali
26	objek_8	Tidak ada kesalahan sama sekali
27	objek_9	Tidak ada kesalahan sama sekali



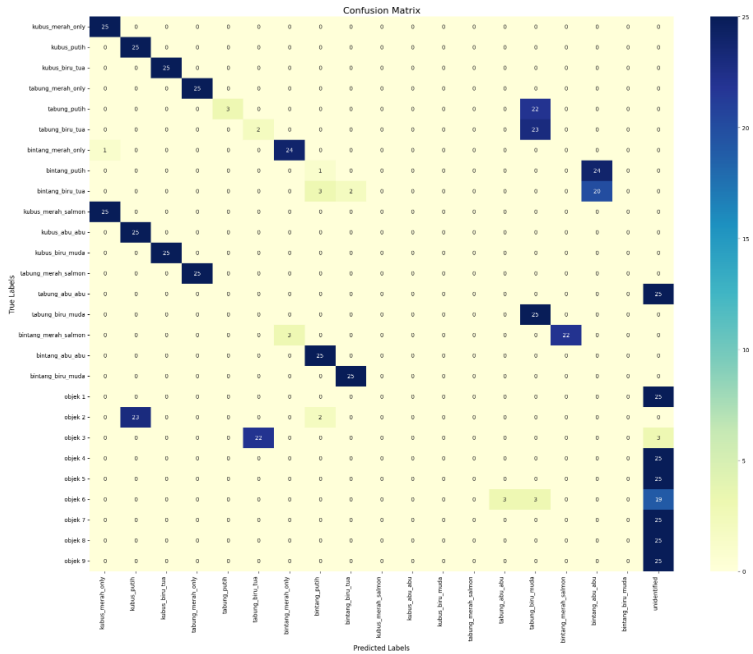
Gambar 21. Model Klasifikasi *Confusion Matrix* pada Hasil Pengujian Deteksi Bentuk dan Warna pada Siang Hari

Berdasarkan hasil klasifikasi dari model *confusion matrix* pada hasil pengujian deteksi bentuk dan warna pada siang hari diatas dapat disimpulkan pada tabel di bawah ini.

Tabel 10. Hasil Klasifikasi Deteksi Bentuk dan Warna pada Siang Hari

No	Nama Objek	Keterangan
1	kubus_merah_only	Tidak ada kesalahan sama sekali
2	kubus_putih	Tidak ada kesalahan sama sekali
3	kubus_biru_tua	Tidak ada kesalahan sama sekali
4	tabung_merah_only	Tidak ada kesalahan sama sekali
5	tabung_putih	Tidak ada kesalahan sama sekali
6	tabung_biru_tua	Tidak ada kesalahan sama sekali
7	bintang_merah_only	Kesalahan deteksi sebagai objek kubus satu kali
8	bintang_putih	Tidak ada kesalahan sama sekali

9	bintang_biru_tua	Tidak ada kesalahan sama sekali
10	kubus_merah_salmon	Kesalahan deteksi berwarna merah 25 kali
11	kubus_abu_abu	Kesalahan deteksi berwarna putih 25 kali
12	kubus_biru_muda	Kesalahan deteksi berwarna biru tua 25 kali
13	tabung_merah_salmon	Kesalahan deteksi berwarna merah 25 kali
14	tabung_abu_abu	Tidak terdeteksi sama sekali
15	tabung_biru_muda	Kesalahan deteksi berwarna biru tua 25 kali
16	bintang_merah_salmon	Kesalahan deteksi sebagai objek berwarna merah tiga kali
17	bintang_abu_abu	Kesalahan deteksi sebagai objek berwarna putih 25 kali
18	bintang_biru_muda	Kesalahan deteksi sebagai objek berwarna biru tua 25 kali
19	objek_1	Tidak ada kesalahan sama sekali
20	objek_2	Kesalahan deteksi sebagai bintang_putih sebanyak 23 kali dan kubus_putih dua kali
21	objek_3	Kesalahan deteksi sebagai tabung_biru_tua sebanyak 22 kali
22	objek_4	Tidak ada kesalahan sama sekali
23	objek_5	Tidak ada kesalahan sama sekali
24	objek_6	Kesalahan deteksi sebagai tabung_biru_muda sebanyak enam kali
25	objek_7	Tidak ada kesalahan sama sekali
26	objek_8	Tidak ada kesalahan sama sekali
27	objek_9	Tidak ada kesalahan sama sekali



Gambar 22. Model Klasifikasi *Confusion Matrix* pada Hasil Pengujian Deteksi Bentuk dan Warna pada Sore Hari

Berdasarkan hasil klasifikasi dari model *confusion matrix* pada hasil pengujian deteksi bentuk dan warna pada sore hari diatas dapat disimpulkan pada tabel di bawah ini.

Tabel 11. Hasil Klasifikasi Deteksi Bentuk dan Warna pada Sore Hari

No	Nama Objek	Keterangan
1	kubus_merah_only	Tidak ada kesalahan sama sekali
2	kubus_putih	Tidak ada kesalahan sama sekali
3	kubus_biru_tua	Tidak ada kesalahan sama sekali
4	tabung_merah_only	Tidak ada kesalahan sama sekali
5	tabung_putih	Kesalahan deteksi sebagai tabung biru muda sebanyak 22 kali
6	tabung_biru_tua	Kesalahan deteksi sebagai tabung biru muda sebanyak 23 kali

7	bintang_merah_only	Kesalahan deteksi sebagai objek kubus satu kali
8	bintang_putih	Kesalahan deteksi berwarna abu abu sebanyak 24 kali
9	bintang_biru_tua	Kesalahan deteksi berwarna abu abu sebanyak 20 kali dan putih tiga kali
10	kubus_merah_salmon	Kesalahan deteksi berwarna merah 25 kali
11	kubus_abu_abu	Kesalahan deteksi berwarna putih 25 kali
12	kubus_biru_muda	Kesalahan deteksi berwarna biru tua 25 kali
13	tabung_merah_salmon	Kesalahan deteksi berwarna merah 25 kali
14	tabung_abu_abu	Tidak terdeteksi sama sekali
15	tabung_biru_muda	Tidak ada kesalahan sama sekali
16	bintang_merah_salmon	Kesalahan deteksi sebagai objek berwarna merah tiga kali
17	bintang_abu_abu	Kesalahan deteksi sebagai objek berwarna putih 25 kali
18	bintang_biru_muda	Kesalahan deteksi sebagai objek berwarna biru tua 25 kali
19	Objek 1	Tidak ada kesalahan sama sekali
20	Objek 2	Kesalahan deteksi sebagai kubus putih sebanyak 23 kali dan bintang putih dua kali
21	Objek 3	Kesalahan deteksi sebagai tabung_biru_tua sebanyak 22 kali
22	Objek 4	Tidak ada kesalahan sama sekali
23	Objek 5	Tidak ada kesalahan sama sekali
24	Objek 6	Kesalahan deteksi sebagai tabung_biru_muda sebanyak tiga kali dan tabung abu abu tiga kali
25	Objek 7	Tidak ada kesalahan sama sekali
26	Objek 8	Tidak ada kesalahan sama sekali
27	Objek 9	Tidak ada kesalahan sama sekali

Setelah memperoleh hasil *confusion matrix*, langkah berikutnya adalah menghitung akurasi dan presisi untuk mengevaluasi kinerja sistem. Akurasi menunjukkan kemampuan sistem mengklasifikasi dengan benar secara keseluruhan, sementara presisi menunjukkan keandalan deteksi positif. Penghitungan ini dilakukan berdasarkan pengujian pada tiga kondisi pencahayaan berbeda. Berikut adalah hasil penghitungan akurasi dan presisi nya.

Tabel 12. Hasil Perhitungan Akurasi dan Presisi

Nama Objek	Akurasi (%)			Presisi (%)		
	Siang	Sore	Ringlight	Siang	Sore	Ringlight
kubus_merah_only	100	94,22	99,37	49,02	49,02	96,15
kubus_putih	100	94,44	100	50	50	100
kubus_biru_tua	100	94,44	100	50	50	100
tabung_merah_only	100	94,44	100	50	50	100
tabung_putih	100	95,11	100	100	100	100
tabung_biru_tua	100	94,89	100	50	100	100
bintang_merah_only	96	99,11	100	88,89	88,89	100
bintang_putih	100	88,44	100	50	3,45	100
bintang_biru_tua	100	89,33	100	50	7,41	100
kubus_merah_salmon	0	94,44	99,16	0	0	92
kubus_abu_abu	0	94,44	100	0	0	100
kubus_biru_muda	0	94,44	100	0	0	100
tabung_merah_salmon	0	94,44	99,37	0	0	100
tabung_abu_abu	0	94,44	99,58	0	0	100
tabung_biru_muda	0	90	100	0	35,71	100
bintang_merah_salmon	88	99,33	100	100	100	100
bintang_abu_abu	0	84,67	98,73	0	0	100
bintang_biru_muda	0	94,44	100	0	0	100

objek_1	100	100	100	100	100	100
objek_2	99,24	99,24	99,24	92	92	92
objek_3	98,86	98,86	98,86	88	88	88
objek_4	100	100	100	100	100	100
objek_5	96,38	96,38	96,38	100	100	100
objek_6	100	100	100	100	100	100
objek_7	100	100	100	100	100	100
objek_8	100	100	100	100	100	100
objek_9	100	100	100	100	100	100
Rata-Rata Total:	69,57	95,54	99,65	56,22	56,09	98,82
Total:	88,26			70,38		

Berdasarkan tabel akurasi dan presisi di atas, pengujian dengan cahaya *ringlight* menunjukkan performa sistem yang sangat baik, dengan rata-rata akurasi 99,65% dan presisi 98,82%. Sebaliknya, pada pengujian dengan cahaya siang hari, rata-rata akurasi dan presisi menurun menjadi 69,57% dan 56,22%, sementara pada cahaya sore hari, akurasi mencapai 95,54%, tetapi presisi hanya 56,09%. Hasil akhir menunjukkan rata-rata akurasi sistem sebesar 88,26% dan presisi 70,38%, yang mengindikasikan bahwa performa sistem sangat bergantung pada intensitas cahaya selama pengujian.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Pada Setelah beberapa pengujian dan pembahasan yang telah dilakukan dapat ditarik kesimpulan jika,

1. Keberhasilan implementasi sistem otomasi sistem sortir berbasis *webcam* dan *PLC Outseal* berhasil dikembangkan untuk menyortir barang berdasarkan warna dan bentuk. Teknologi otomasi ini menggantikan metode manual dengan efisiensi yang lebih tinggi.
2. Tingkat akurasi dan pengaruh pencahayaan pada algoritma *YOLO v5* yang digunakan menunjukkan tingkat akurasi sebesar 88,26% dan presisi 70,38%, terutama dalam kondisi pencahayaan optimal. Pencahayaan tambahan seperti *ringlight* terbukti meningkatkan akurasi deteksi bentuk dan warna secara *real-time*.
3. Integrasi komponen yang berhasil komponen utama seperti conveyor, motor *servo*, sensor *proximity*, dan *PLC Outseal* berhasil diintegrasikan dengan baik, memungkinkan proses sortir berjalan otomatis dan konsisten, sehingga mampu mengurangi waktu serta biaya dalam penyortiran barang.

5.2. Saran

Adapun beberapa saran terhadap pengembangan dari alat ini yaitu,

1. Penggunaan sensor tambahan atau pengaturan pencahayaan yang lebih optimal dapat mendukung akurasi deteksi pada objek dengan permukaan reflektif atau warna gelap.
2. Meningkatkan kapasitas *conveyor* agar mampu mendeteksi beberapa objek dalam satu frame dapat mempercepat waktu sortir dan menambah efisiensi sistem.
3. Algoritma *YOLO v5* dapat disempurnakan dengan menguji varian yang lebih baru atau memadukan metode lain untuk meningkatkan kecepatan pemrosesan dan ketepatan hasil deteksi.

Daftar Pustaka

- [1] R. Bangun *et al.*, "JTEV (JURNAL TEKNIK ELEKTRO DAN VOKASIONAL)", [Online]. Available: <http://ejournal.unp.ac.id/index.php/jtev/index>
- [2] R. H. Hutabarat, S. R. Sulistyanti, and E. Nasrullah, "Rancang bangun konveyor penyortiran barang dengan pengenalan pola bentuk dan warna menggunakan webcam," *The Electrician*, vol. 7, no. 2, pp. 73–77, 2013.
- [3] B. Gemilang, L. Nurpulaela, and Y. Saragih, "Implementasi Outseal PLC Pada Automatic Duck Egg Washing Machine," *MULTINETICS*, vol. 6, no. 2, pp. 117–127, Dec. 2020, doi: 10.32722/multinetics.v6i2.3054.
- [4] F. E. Ahmad and E. Fitriani, "Penggunaan sistem Outseal PLC pada pemilah otomatis dan penghitung otomatis," *Bina Darma Conference on Engineering Science (BDCES)*, vol. 2, no. 2, pp. 27–39, 2020.
- [5] H. Singgih, "Uji kandungan formalin pada ikan asin menggunakan sensor warna dengan bantuan FMR (Formalin Main Reagent)," *ELTEK*, vol. 11, no. 1, pp. 55–70, 2017.
- [6] H. Singgih, "UJI KANDUNGAN FORMALIN PADA IKAN ASIN MENGGUNAKAN SENSOR WARNA DENGAN BANTUAN FMR (Formalin Main Reagent)."
- [7] B. Priyadi, "APLIKASI SENSOR WARNA JENIS TCS 230 SEBAGAI ALAT PENENTU KOMPOSISI WARNA PADA CAT MOBIL."
- [8] Ö. G. Saracoglu and H. Altural, "Color regeneration from reflective color sensor using an artificial intelligent technique," *Sensors*, vol. 10, no. 9, pp. 8363–8374, Sep. 2010, doi: 10.3390/s100908363.
- [9] Kata, "DAFTAR ISI."
- [10] Y. Arvio, D. T. Kusuma, and I. B. Sangadji, "PENDEKATAN ALGORITMA YOLO V5 UNTUK MENDETEKSI CACAT PRODUK MASKER," *Dinamika Rekayasa*, vol. 20, no. 1, pp. 11–17, Jan. 2024, doi: 10.20884/1.dinarek.2024.20.1.33.
- [11] "NEMA 23 Stepper Motor," *Components101*. [Online]. Available: <https://components101.com/motors/nema-23-stepper-motor-datasheet-specs>.
- [12] Administrator, "Apa itu motor stepper: Pengertian, cara kerja dan kelebihanannya," *Empat Pilar*, May 1, 2023. [Online]. Available: <https://www.empatpilar.com/apa-itu-motor-stepper/>.
- [13] De Bakker, "TB6600 Stepper Motor Driver with Arduino Tutorial," *Makerguides.com*, Jan. 29, 2024. [Online]. Available: <https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/>.
- [14] G. W. Pambudi and G. W. Pambudi, "Cara kontrol motor stepper dengan driver TB6600 / TB6500 Arduino," *Cronyos*, May 16, 2020. [Online]. Available: <https://www.cronyos.com/cara-kontrol-motor-stepper-dengan-driver-tb6600-tb6500-arduino/>.

- [15] "TB6600 Stepper Motor Driver Module," Components101. [Online]. Available: <https://components101.com/modules/tb6600-stepper-motor-driver-module-pinout-features-datasheet-working-application-alternative>.
- [16] Media, "Cara kerja sensor photoelectric," Pak Guru, Aug. 29, 2021. [Online]. Available: <https://pakguru.co.id/cara-kerja-sensor-photoelectric/>.
- [17] Litalia, "Pengertian webcam, fungsi webcam dan cara kerja," JurnalPonsel, Dec. 9, 2023. [Online]. Available: <https://www.jurnalponcel.com/pengertian-webcam-fungsi-webcam-dan-cara-kerja/>.
- [18] S. Yulianti, "Pengertian dan penggunaan Outseal PLC pada industri," IndoNiaga, Jan. 31, 2022. [Online]. Available: <https://www.indoniaga.co.id/2022/01/pengertian-dan-penggunaan-outseal-plc.html>.
- [19] R. Gargeya, "Understanding confusion matrix," Towards Data Science, Dec. 12, 2018. [Online]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

Biodata



Nama : Shendy Saputra
TTL : Padang, 22 Mei 2003
Agama : Islam
Alamat : Mangsang Permai Blok M No 30

Email : shendysaputra677@gmail.com
Riwayat Pendidikan SMA/SMK: SMK Negeri 3 Batam
SMP : SMP Negeri 40 Batam



Nama : Anastasya Andaresta Pelmelay
TTL : Batam, 20 Juni 2001
Agama : Kristen
Alamat : Permata Hijau Blok F No. 4

Email : tsyplmly@gmail.com
Riwayat Pendidikan SMA/SMK: SMK Negeri 1 Batam
SMP : SMP Negeri 9 Batam

Lampiran

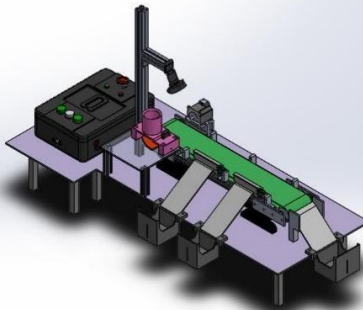
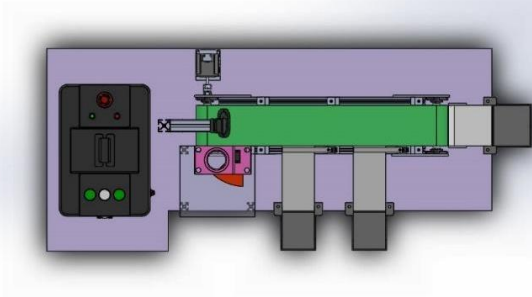
Lampiran A. Hasil Perancangan Mekanikal



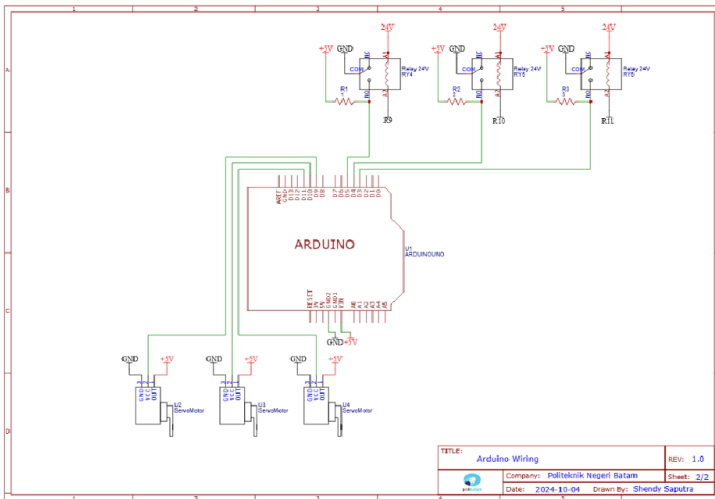
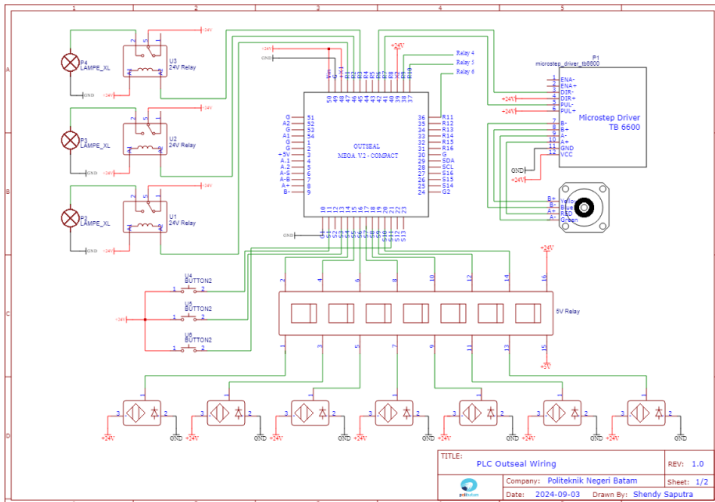
Lampiran B. Hasil Perancangan Elektrikal



Lampiran C. Perancangan Mekanikal



Lampiran D. Schematic Elektrikal



Lampiran E. Hasil Deteksi Objek pada GUI

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



kubus_abu_abu 0.89

TUGAS AKHIR
Sistem Sortir Barang
Berdasarkan Warna dan Bentuk
Menggunakan Webcam
Berdasarkan PLC Outseal

Shandy Saputra_A2121003
Akhilanya Andaranta Pelmisay_A2121008

START

STOP

STATUS WARNA
Abu Abu: Terdeteksi

STATUS BENTUK
Kubus: Terdeteksi

Data Objek

Camera 1

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STOP

STATUS WARNA

Biru Tua: Terdeteksi

Data Objek

Camera 1

STATUS BENTUK

Kubus: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Merah Salmon: Terdeteksi

Camera 1

STATUS BENTUK

Kubus: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STOP

STATUS WARNA

Biru Muda: Terdeteksi

STATUS BENTUK

Kubus: Terdeteksi

Data Objek

Camera 1

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

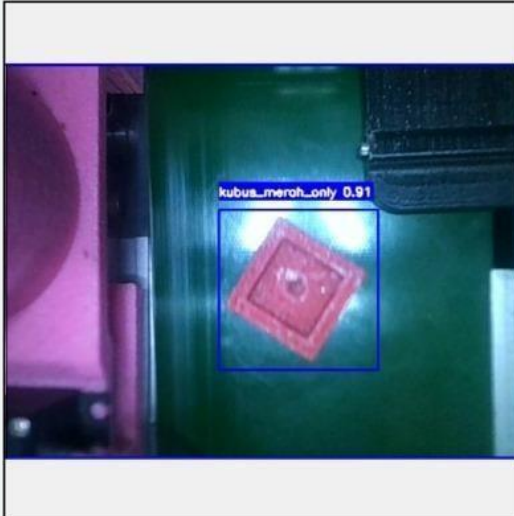
Putih: Terdeteksi

Camera 1 →

STATUS BENTUK

Kubus: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START	STATUS WARNA Merah: Terdeteksi	Data Objek
STOP	STATUS BENTUK Kubus: Terdeteksi	Camera 1

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STOP

STATUS WARNA

Merah: Terdeteksi

STATUS BENTUK

Tabung: Terdeteksi

Data Objek

Camera 1

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Abu /Abu: Terdeteksi

Camera 1 →

STATUS BENTUK

Tabung: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Biru Tua: Terdeteksi

Camera 1

STATUS BENTUK

Tabung: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STOP

STATUS WARNA

Putih: Terdeteksi

STATUS BENTUK

Tabung: Terdeteksi

Data Objek

Camera 1

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Binu Muda: Terdeteksi

Camera 1

STATUS BENTUK

Tabung: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA
Merah Salmon: Terdeteksi

Data Objek

STOP

Camera 1

STATUS BENTUK
Tabung: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA
Biru Tua: Terdeteksi

Data Objek

STOP

Camera 1

STATUS BENTUK
Bintang: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Biru Muda: Terdeteksi

Camera 1

STATUS BENTUK

Bintang: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

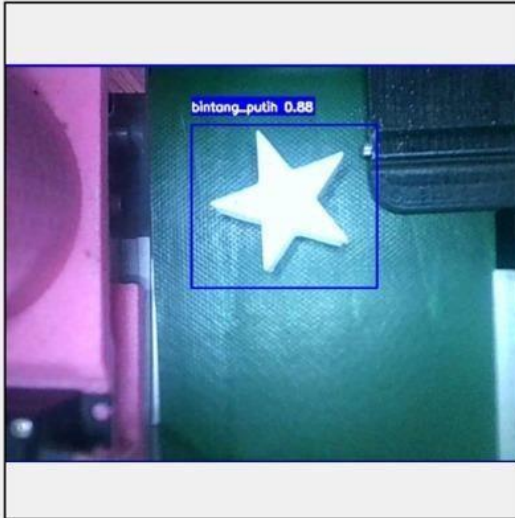
Merah Salmon: Terdeteksi

Camera 1

STATUS BENTUK

Bintang: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Putih: Terdeteksi

Camera 1

STATUS BENTUK

Bintang: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Merah: Terdeteksi

Camera 1

STATUS BENTUK

Bintang: Terdeteksi

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STOP

STATUS WARNA

Abu Abu: Terdeteksi

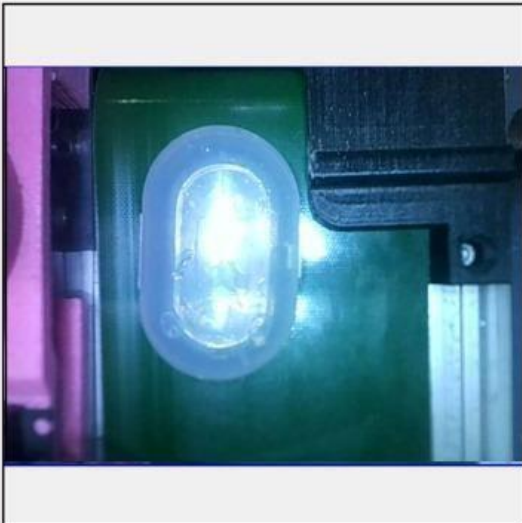
STATUS BENTUK

Bintang: Terdeteksi

Data Objek

Camera 1 →

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

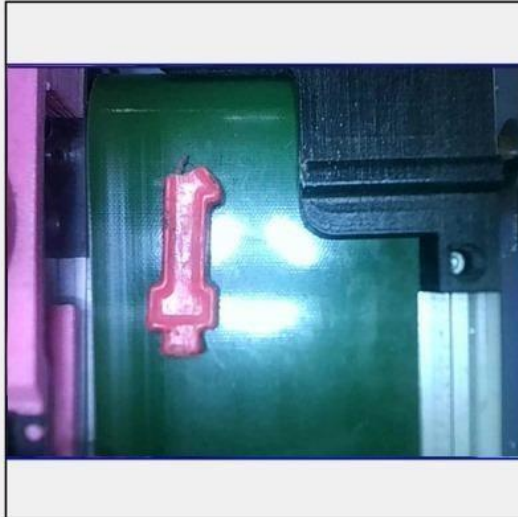
Data Objek

STOP

Camera 1

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Camera 1 →

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

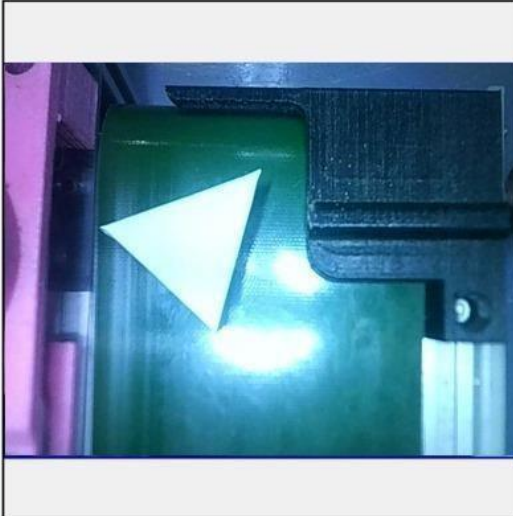
Data Objek

STOP

Camera 1 →

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Camera 1

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

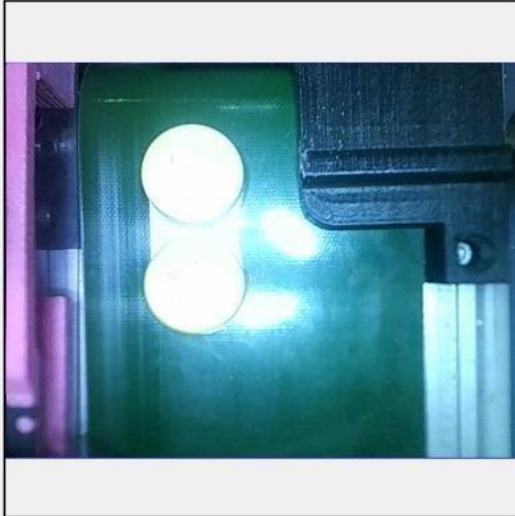
Data Objek

STOP

Camera 1

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

STATUS BENTUK

Camera 1 

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Camera 1

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

Data Objek

STOP

Camera 1

STATUS BENTUK

SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK



START

STATUS WARNA

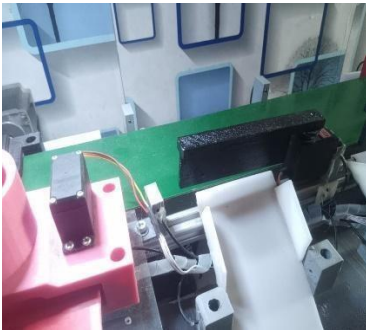
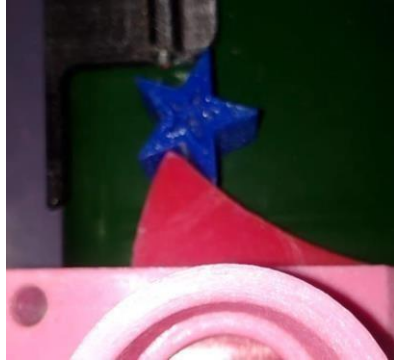
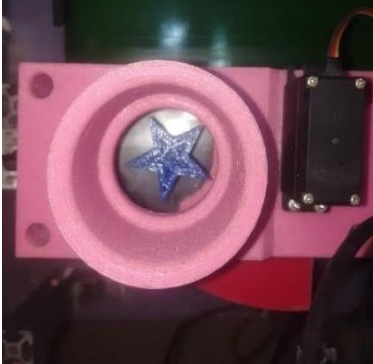
Data Objek

STOP

Camera 1

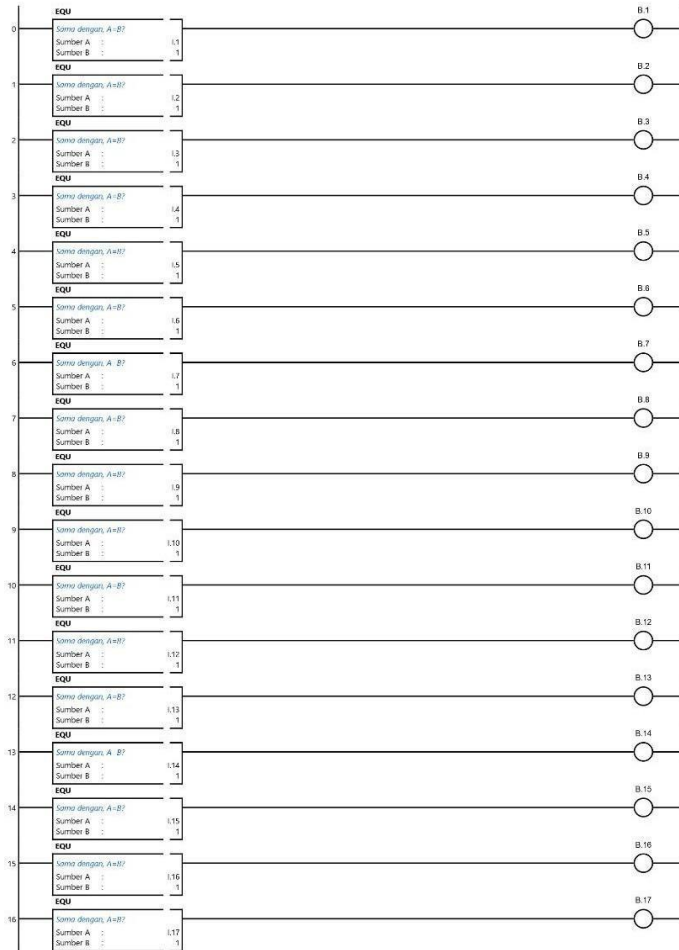
STATUS BENTUK

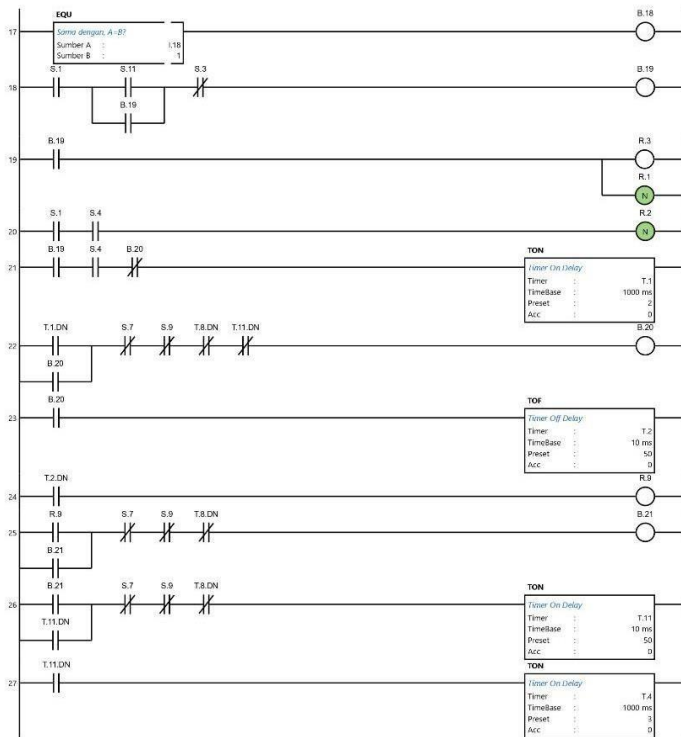
Lampiran D. Percobaan Deteksi Adanya Benda

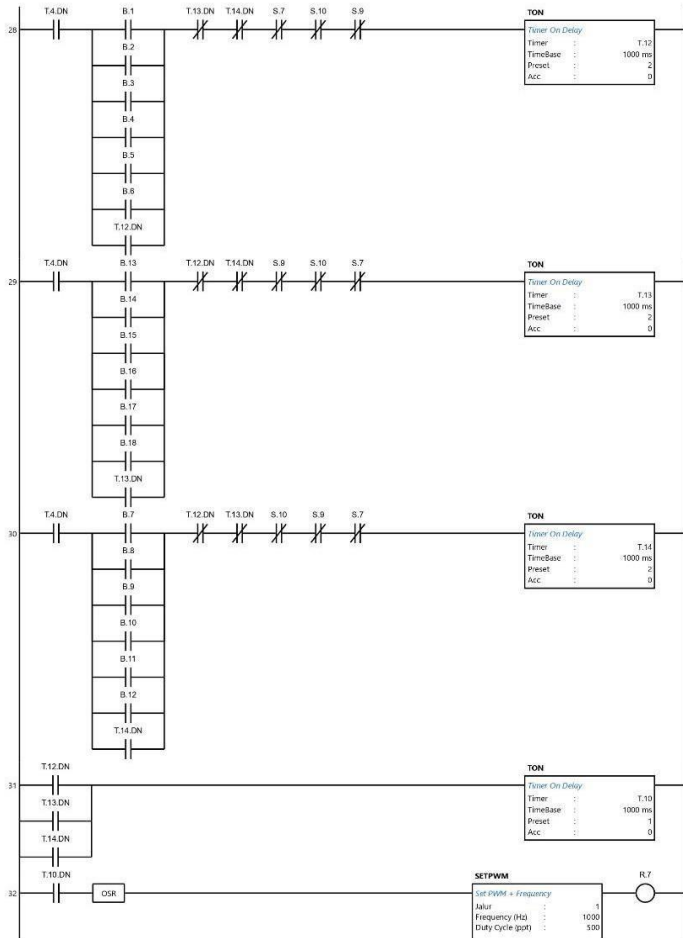


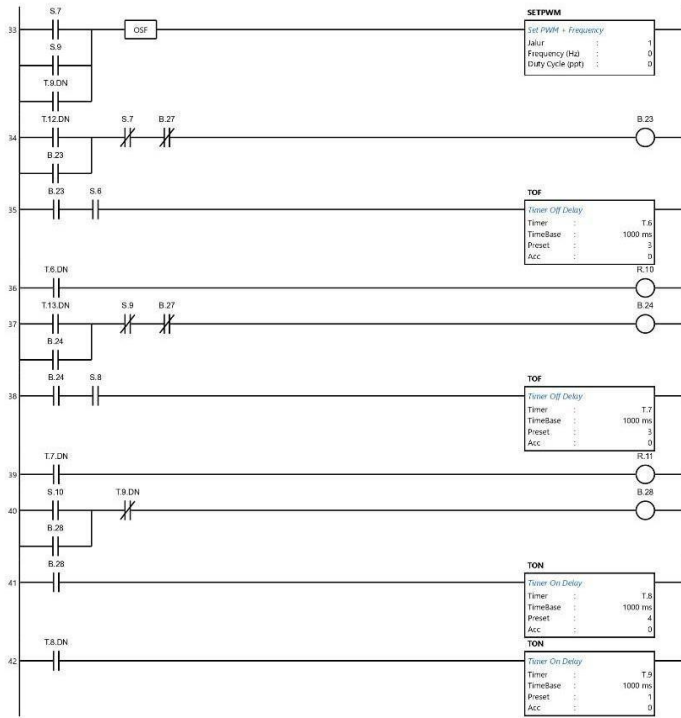
Lampiran F. Program Sistem Kendali PLC

Judul : TA BISSMILAH, Main
 Halaman : 1 Catatan :









Lampiran G. Program Komunikasi dan UI

```
import tkinter as tk
import tkinter.font as tkFont
from tkinter import filedialog
import cv2
import minimalmodbus
from PIL import Image, ImageTk
import numpy as np

INPUT_SIZE = 640
CONFIDENCE_THRESHOLD = 0.45
NMS_THRESHOLD = 0.45

def iou(box1, box2):
    x1, y1, w1, h1 = box1[:4]
    x2, y2, w2, h2 = box2[:4]

    intersection_left = max(x1, x2)
    intersection_top = max(y1, y2)
    intersection_right = min(x1 + w1, x2 + w2)
    intersection_bottom = min(y1 + h1, y2 + h2)

    intersection_area = max(0, intersection_right - intersection_left) * max(0,
intersection_bottom - intersection_top)

    box1_area = w1 * h1
    box2_area = w2 * h2

    union_area = box1_area + box2_area - intersection_area

    iou_value = intersection_area / union_area

    return iou_value

class App:
    def __init__(self, root):
        self.root = root
        root.title("SISTEM SORTIR BERDASARKAN WARNA DAN BENTUK")
        width = 750
        height = 820
```

```

# Declare PLC
self.instrument = minimalmodbus.Instrument('COM8', 1)
self.instrument.serial.baudrate = 115200

# Load the logo image
logo_image = Image.open("logo_poltek.jpg")
logo_image = logo_image.resize((180, 503),
Image.Resampling.LANCZOS)
self.logo_photo = ImageTk.PhotoImage(logo_image)
self.logo_label = tk.Label(root, image=self.logo_photo)
self.logo_label.place(x=550, y=105)

# Create a frame for the headline
headline_frame = tk.Frame(root, bg="white")
headline_frame.pack(side=tk.TOP, fill=tk.X)

# Create a label for the headline
headline_label = tk.Label(headline_frame, text="SISTEM SORTIR
BERDASARKAN WARNA DAN BENTUK",
font=("Arial", 16, "bold"), bg="white")
headline_label.pack(side=tk.LEFT, padx=20, pady=10)

screenwidth = root.winfo_screenwidth()
screenheight = root.winfo_screenheight()
alignstr = '%dx%d+%d+%d' % ((width, height) +
((screenwidth - width) // 2, (screenheight - height) // 2))
root.geometry(alignstr)
root.resizable(width=True, height=True)

self.capture = None
self.classes =
['bintang_abu_abu','bintang_biru_muda','bintang_biru_tua','bintang_merah_onl
y',
'bintang_merah_salmon','bintang_putih','kubus_abu_abu','kubus_biru_muda',
'kubus_biru_tua','kubus_merah_only','kubus_merah_salmon','kubus_putih',
'tabung_abu_abu','tabung_biru_muda','tabung_biru_tua','tabung_merah_only',
'tabung_merah_salmon','tabung_putih']

# Shape Status Frame

```

```

shape_frame = tk.Frame(root, bd=2, relief="solid")
shape_frame.place(x=190, y=720, width=185, height=60)

shape_title = tk.Label(shape_frame, text="STATUS BENTUK",
font=("Arial", 11, "bold"))
shape_title.pack(pady=5)

# Buat frame kosong untuk shape status
self.shape_status_frame = tk.Frame(shape_frame)
self.shape_status_frame.pack(fill='x', padx=5)

# Color Status Frame
color_frame = tk.Frame(root, bd=2, relief="solid")
color_frame.place(x=190, y=650, width=185, height=60)

color_title = tk.Label(color_frame, text="STATUS WARNA", font=("Arial",
11, "bold"))
color_title.pack(pady=5)

# Buat frame kosong untuk color status
self.color_status_frame = tk.Frame(color_frame)
self.color_status_frame.pack(fill='x', padx=5)

# Dictionary untuk menyimpan status aktif
self.active_shapes = set()
self.active_colors = set()

self.selected_camera = tk.StringVar(root)
self.cameras = self.get_available_cameras()
self.selected_camera.set(self.cameras[0]) if self.cameras else
self.selected_camera.set("No Camera")

GListBox_128 = tk.OptionMenu(root, self.selected_camera,
*self.cameras)
GListBox_128.config(bd=2, relief="solid")
GListBox_128.place(x=405, y=690, width=120, height=25)

self.canvas = tk.Canvas(root, width=500, height=500, bd=2,
relief="solid")
self.canvas.place(x=20, y=105)

```

```

        GButton_536 = tk.Button(root, text="START",
command=self.turn_on_camera, bd=2, relief="solid")
        GButton_536.place(x=21, y=650, width=140, height=25)

        GButton_599 = tk.Button(root, text="STOP",
command=self.turn_off_camera, bd=2, relief="solid")
        GButton_599.place(x=21, y=690, width=140, height=25)

        GButton_load = tk.Button(root, text="Data Objek",
command=self.load_file, bd=2, relief="solid")
        GButton_load.place(x=405, y=650, width=120, height=25)

def get_available_cameras(self):
    cameras = []
    for i in range(10):
        cap = cv2.VideoCapture(i)
        if cap.isOpened():
            cameras.append(f"Camera {i}")
            cap.release()
    return cameras

def turn_off_camera(self):
    if self.capture is not None:
        self.capture.release()
        self.capture = None
        self.canvas.delete("all")
        # Clear status frames
        for widget in self.shape_status_frame.winfo_children():
            widget.destroy()
        for widget in self.color_status_frame.winfo_children():
            widget.destroy()

def turn_on_camera(self):
    if self.capture is None:
        camera_index = int(self.selected_camera.get().split()[-1])
        self.capture = cv2.VideoCapture(camera_index)
        self.update()

def load_file(self):
    file_path = filedialog.askopenfilename(filetypes=[("ONNX Files",
"**.onnx")])
    if file_path:

```

```

self.net = cv2.dnn.readNetFromONNX(file_path)
print(f"Loaded ONNX file: {file_path}")

def update(self):
    if self.capture is not None:
        ret, frame = self.capture.read()
        if ret:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            img_height, img_width = frame.shape[:2]
            x_scale = img_width / INPUT_SIZE
            y_scale = img_height / INPUT_SIZE
            detections = self.detect_objects(frame, self.classes, self.net, x_scale,
y_scale)

            # Reset status sets
            self.active_shapes.clear()
            self.active_colors.clear()

            # Hapus semua label status yang ada
            for widget in self.shape_status_frame.winfo_children():
                widget.destroy()
            for widget in self.color_status_frame.winfo_children():
                widget.destroy()

            for detection in detections:
                x1, y1, width, height, detected_class, confidence = detection
                x2 = x1 + width
                y2 = y1 + height
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)

                label = detected_class
                conf = confidence
                text = f"{label} {conf:.2f}"

            # Update shape status
            if 'bintang' in detected_class and 'Bintang' not in
self.active_shapes:
                self.active_shapes.add('Bintang')
                tk.Label(self.shape_status_frame,
                    text="Bintang: Terdeteksi",
                    fg="green",
                    font=("Arial", 9)).pack(pady=2)

```

```

        elif 'kubus' in detected_class and 'Kubus' not in
self.active_shapes:
        self.active_shapes.add('Kubus')
        tk.Label(self.shape_status_frame,
            text="Kubus: Terdeteksi",
            fg="green",
            font=("Arial", 9)).pack(pady=2)

        elif 'tabung' in detected_class and 'Tabung' not in
self.active_shapes:
        self.active_shapes.add('Tabung')
        tk.Label(self.shape_status_frame,
            text="Tabung: Terdeteksi",
            fg="green",
            font=("Arial", 9)).pack(pady=2)

        # Update color status
        if 'abu_abu' in detected_class and 'Abu-abu' not in
self.active_colors:
        self.active_colors.add('Abu-abu')
        tk.Label(self.color_status_frame,
            text="Abu-abu: Terdeteksi",
            fg="green",
            font=("Arial", 9)).pack(pady=2)

        elif 'biru_muda' in detected_class and 'Biru Muda' not in
self.active_colors:
        self.active_colors.add('Biru Muda')
        tk.Label(self.color_status_frame,
            text="Biru Muda: Terdeteksi",
            fg="green",
            font=("Arial", 9)).pack(pady=2)

        elif 'biru_tua' in detected_class and 'Biru Tua' not in
self.active_colors:
        self.active_colors.add('Biru Tua')
        tk.Label(self.color_status_frame,
            text="Biru Tua: Terdeteksi",
            fg="green",
            font=("Arial", 9)).pack(pady=2)

```

```

        elif 'merah_only' in detected_class and 'Merah' not in
self.active_colors:
            self.active_colors.add('Merah')
            tk.Label(self.color_status_frame,
                    text="Merah: Terdeteksi",
                    fg="green",
                    font=("Arial", 9)).pack(pady=2)

        elif 'merah_salmon' in detected_class and 'Merah Salmon' not in
self.active_colors:
            self.active_colors.add('Merah Salmon')
            tk.Label(self.color_status_frame,
                    text="Merah Salmon: Terdeteksi",
                    fg="green",
                    font=("Arial", 9)).pack(pady=2)

        elif 'putih' in detected_class and 'Putih' not in self.active_colors:
            self.active_colors.add('Putih')
            tk.Label(self.color_status_frame,
                    text="Putih: Terdeteksi",
                    fg="green",
                    font=("Arial", 9)).pack(pady=2)

        # Calculate text position and draw it
        text_size, _ = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX,
0.5, 2)

        text_width, text_height = text_size
        text_x = x1
        text_y = y1 - text_height - 5 if y1 - text_height - 5 > 0 else y1 + 5

        cv2.rectangle(frame, (text_x, text_y - text_height - 5),
                    (text_x + text_width, text_y + 5), (0, 0, 255), -1)
        cv2.putText(frame, text, (text_x, text_y),
cv2.FONT_HERSHEY_SIMPLEX,
                    0.5, (255, 255, 255), 2)

        try:
            # Handle PLC communication based on detected class
            if detected_class in self.classes:
                index = self.classes.index(detected_class)
                # Set detected class register to 1 and others to 0
                for i in range(len(self.classes)):

```

```

        value = 1 if i == index else 0
        self.instrument.write_register(i, value, 0, 6)
    else:
        # Set all registers to 0 if no valid class detected
        for i in range(len(self.classes)):
            self.instrument.write_register(i, 0, 0, 6)

    except minimalmodbus.NoResponseError as e:
        print(f"Error there is no response from PLC : {e}")

    # Get canvas dimensions and resize frame
    canvas_width = self.canvas.wininfo_width()
    canvas_height = self.canvas.wininfo_height()

    frame_height, frame_width = frame.shape[:2]
    aspect_ratio = frame_width / frame_height

    if canvas_width / canvas_height > aspect_ratio:
        new_height = canvas_height
        new_width = int(new_height * aspect_ratio)
    else:
        new_width = canvas_width
        new_height = int(new_width / aspect_ratio)

    frame = cv2.resize(frame, (new_width, new_height))
    frame = cv2.copyMakeBorder(frame, 2, 2, 2, 2,
cv2.BORDER_CONSTANT, value=(0, 0, 255))

    img = Image.fromarray(frame)
    imgtk = ImageTk.PhotoImage(image=img)
    self.canvas.imgtk = imgtk
    self.canvas.create_image((canvas_width - new_width) // 2,
        (canvas_height - new_height) // 2,
        anchor=tk.NW, image=imgtk)

    self.root.after(10, self.update)

    def detect_objects(self, frame, classes, net, x_scale, y_scale):
        blob = cv2.dnn.blobFromImage(frame, scalefactor=1/255,
size=[INPUT_SIZE, INPUT_SIZE],
            mean=[0, 0, 0], swapRB=False, crop=False)
        net.setInput(blob)

```

```

detections_raw = net.forward()[0]

classes_ids = []
confidences = []
boxes = []
rows = detections_raw.shape[0]

for i in range(rows):
    row = detections_raw[i]
    confidence = row[4]
    if confidence > CONFIDENCE_THRESHOLD:
        classes_score = row[5:]
        ind = np.argmax(classes_score)
        if classes_score[ind] > CONFIDENCE_THRESHOLD:
            classes_ids.append(ind)
            confidences.append(confidence)
            cx, cy, w, h = row[:4]
            x1 = int((cx - w / 2) * x_scale)
            y1 = int((cy - h / 2) * y_scale)
            width = int(w * x_scale)
            height = int(h * y_scale)
            box = np.array([x1, y1, width, height])
            boxes.append(box)

indices = cv2.dnn.NMSBoxes(boxes, confidences, NMS_THRESHOLD,
NMS_THRESHOLD)

detections = []
for i in indices:
    x1, y1, w, h = boxes[i]
    label = classes[classes_ids[i]]
    conf = confidences[i]
    detections.append((x1, y1, w, h, label, conf))

return detections

def apply_nms(self, detections):
    detections = sorted(detections, key=lambda x: x[5], reverse=True)
    nms_detections = []
    while detections:
        current = detections.pop(0)
        nms_detections.append(current)

```

```

        for detection in detections[:]:
            if iou(current, detection) > NMS_THRESHOLD:
                detections.remove(detection)

    return nms_detections

if __name__ == "__main__":
    root = tk.Tk()
    app = App(root)
    root.mainloop()

```

Lampiran H. Program Kendali Servo

```

#include <Servo.h>

// Pin tombol
const int tombol1Pin = 3;
const int tombol2Pin = 4;
const int tombol3Pin = 5;

// Servo
Servo servo1;
Servo servo2;
Servo servo3;

// Posisi servo
const int servo1Default = 75;
const int servo1Pressed = 0;
const int servo2Default = 130;
const int servo2Pressed = 180;
const int servo3Default = 130;
const int servo3Pressed = 180;

void setup() {
    // Attach servo ke pin PWM
    servo1.attach(9);
    servo2.attach(10);
    servo3.attach(11);

    // Atur posisi awal servo

```

```

servo1.write(servo1Default);
servo2.write(servo2Default);
servo3.write(servo3Default);

// Atur pin tombol sebagai input pull-up
pinMode(tombol1Pin, INPUT_PULLUP);
pinMode(tombol2Pin, INPUT_PULLUP);
pinMode(tombol3Pin, INPUT_PULLUP);
}

void loop() {
// Membaca status tombol (active LOW)
bool tombol1State = digitalRead(tombol1Pin) == LOW;
bool tombol2State = digitalRead(tombol2Pin) == LOW;
bool tombol3State = digitalRead(tombol3Pin) == LOW;

// Kontrol servo berdasarkan status tombol
if (tombol1State) {
servo1.write(servo1Pressed);
} else {
servo1.write(servo1Default);
}

if (tombol2State) {
servo2.write(servo2Pressed);
} else {
servo2.write(servo2Default);
}

if (tombol3State) {
servo3.write(servo3Pressed);
} else {
servo3.write(servo3Default);
}

// Delay untuk debounce tombol
delay(50);
}

```