Sistem Kendali PLC Pada Mesin Pemilahan Objek Berbasis Kamera

Tugas Akhir

Oleh:
Kurniawan Teguh Setia Fatoni (4212001008)

Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2022

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "*Camera Based Smart Sorting on Production Line*" 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,26 Maret 2024

Putri Lidiya
NIM: 4212011010

Aditya Muhammad Kinandhika
NIM: 4212011008

Kurniawan Teguh Setia Fatoni
NIM: 4212001008

Lembar Pengesahan

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

Oleh:
Kurniawan Teguh Setia Fatoni (4212001008)
Aditya Muhammad Kinandhika (4212011008)
Putri Lidiya (4212011010)

Tanggal Sidang: DD MM, YYYY

Disetujui oleh :

1. Widya Rika Puspita, S.Pd., M.Si., Ph.D
NIK: 119211

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

1. Eko Rudiawan Jamzuri,S.ST
NIK: 113117

Camera Based Smart Sorting on Production Line

Abstrak

Perkembangan teknologi sekarang sudah jauh berkembang pesat. Pada proses pemilahan yang dilakukan secara manual telah mulai diganti oleh mesin pemilah otomatis. Penelitian ini bertujuan untuk membuat prototipe pemilahan objek benda secara otomatis menggunakan kamera ELP8 sebagai sensor utama untuk proses deteksi setiap objek benda dengan menggunakan metode YoloV3. YoloV3 akan melakukan pemisahan gambar yang akan dibagi menjadi *grid,* serta melakukan proses *labeling* untuk mendapatkan *dataset* program yang akan di masukkan ke dalam pemrograman *Python.* Data tersebut di simpan sesuai dengan 18 objek yaitu *Box_Blue, Box_Light_Blue, Box_Pink, Box_Red, Box_Silver, Box_White, Circle_Blue, Circle_Light_Blue, Circle_Pink, Circle _Red, Circle _Silver, Circle _White, Star_Blue, Star_Light_Blue, Star _Pink, Star _Red, Star _Silver, Star _White.* Lalu data yang telah di dapat akan dihubungkan melalui Modbus RTU (*Remote Terminal Unit*) menuju PLC (*Programmable Logic Controller*) *Outseal* untuk mengontrol konveyor dan menggerakkan *Servo* yang membantu dalam proses pemilahan menuju ke *Box* pemilahan tiap-tiap objek benda. Hasil percobaan yang dilakukan dengan total 900 kali pengujian, dengan masing-masing 50 kali pengujian pada setiap objek benda. Dengan melakukan pengujian diperoleh nilai terendah dari akurasi 99% dan presisi 84%, menunjukkan bahwa *Camera Based Smart Sorting on Production Line* berhasil mendeteksi objek benda yang memiliki nilai akurasi dan presisi diatas 80% dan mampu melakukan pemilahan untuk objek benda.

Kata kunci : YoloV3, PLC (*Programmable Logic Controller*) *Outseal*, konveyor.

Camera Based Smart Sorting on Production Line

Abstract

The development of technology is now far advanced. The manual sorting process has begun to be replaced by an automatic sorting machine. This research aims to make a prototype of sorting objects automatically using the ELP8 camera as the main sensor for the detection process of each object by using the YoloV3. YoloV3 will perform image separation which will be devided into grids, and perform a labeling process to get a program dataset that will be entered into Python programming. The data is stored according to 18 objects, namely Box_Blue, Box_Light_Blue, Box_Pink, Box_Red, Box_Silver, Box_White, Circle_Blue, Circle_Light_Blue, Circle_Pink, Circle _Red, Circle _Silver, Circle _White, Star_Blue, Star_Light_Blue, Star _Pink, Star _Red, Star _Silver, Star _White. Then the dataset that has been obtained will be connected via Modbus RTU (*Remote Terminal Unit*) *to the Outseal PLC* (*Programmable Logic Controller*) *to control the conveyor and drive the Servo that helps in the sorting process to the sorting box for each object. The results of the experiments carried out with a total of 900 tests, with each 50 tests on each object. By testing the lowest value of 99% accuracy and 84% precision, it shows that Camera Based Smart Sorting on Production Line successfully detects objects that have accuracy and precision values above 80% and is able to perform sorting for objects.*

Keywords: YoloV3, PLC (Programmable Logic Controller) Outseal, conveyor.

Kata Pengantar

Alhamdulillahhirabbil'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 *"Camera Based Smart Sorting on Production Line".* 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 Uuf Brajawidagda, S.T., M.T., Ph.D. selaku Direktur Politeknik Negeri Batam.
3. Bapak Dr.Budi Sugandi, S.T., M.Eng. selaku ketua Jurusan Teknik Elekto Politeknik Negeri Batam.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng. selaku ketua Program Studi Mekatronika Politeknik Negeri Batam.
5. Bapak Eko Rudiawan Jamzuri, S.ST. selaku dosen pembimbing yang telah menyediakan waktu dan pikiran untuk mengarahkan penulis dalam menyelesaikan Tugas Akhir.
6. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah Tugas Akhir.
7. Bapak Sumatri Kurniawan Risandriya, S.T., M.T. selaku wali dosen kelas A pagi dan Dr.Bapak Budi Sugandi, S.T., M.Eng. selaku wali dosen kelas A malam.
8. Ibu Widya Rika Puspita, S.Pd., M.Si., Ph.D dan Bapak Muhammad Naufal Airlanggan Diputra, S.Pd., M.P.H. selaku dosen penguji.
9. Seluruh dosen-dosen Jurusan Teknik Elektro Politeknik Negeri Batam.
10. 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. Semoga Laporan Tugas Akhir ini dapat bermanfaat dan berguna bagi kita semua. Terima kasih atas segala perhatiannya.

Batam, 26 Maret 2024

Kurniawan Teguh Setia Fatoni
Muhammad Aditya Kinandhika
Putri Lidiya

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab 1. Pendahuluan

1.1. Latar Belakang

Dalam industri, produk sering kali diolah dari bahan mentah yang memiliki variasi warna atau bentuk yang berbeda-beda. Sebagai contoh, pada industri otomotif dan tekstil yang membuat warna sebagai tolak ukur kualitas, hal ini sangat penting untuk menentukan kualitas produk akhir. Namun, pengecekan warna secara manual oleh operator manusia sangat sulit dan tidak akurat.[1]

Dalam hal ini, *Camera Based Smart Sorting on Production Line* dapat menjadi solusi yang efektif. *Camera Based Smart Sorting on Production Line* memproses pemilahan suatu objek benda yang menggunakan kamera sebagai pendeteksi serta objek benda akan berjalan melalui konveyor kemudian akan dipisahkan menggunakan *Servo* sebagai alat pemilahnya.

Dengan begitu dalam tugas akhir ini *Camera Based Smart Sorting on Production Line* dapat membantu mempermudah pekerjaan dalam memilah suatu barang yang berbeda warna dan bentuk dengan secara otomatis, jika di lakukan secara manual akan terjadi nya beberapa kesalahan dalam memilah warna dan bentuk, maka *Camera Based Smart Sorting on Production Line* sangat efektif untuk mencegah terjadinya kesalahan dalam pemilahan warna dan bentuk.

Pada tugas akhir ini akan ditambahkan beberapa sistem seperti konveyor, komunikasi Modbus, dan penambahan program untuk mendeteksi bentuk, dikarenakan belum adanya sistem ini pada PBL (*Project Based Learning*) sebelumnya. Penambahan sistem ini memiliki tujuan untuk memberikan kemudahan dalam proses pemilahan yang dilakukan. Sistem konveyor ini juga akan membantu dalam mengatur dan memindahkan objek secara efisien.

Dalam menjalakan fungsinya, alat ini melibatkan penggunaan kamera sebagai pengenal objek dan penentu warna objek melalui teknologi pemrosesan citra yang menggunakan pemrograman *Python*. Kemudian, data hasil pengolahan citra dapat dikirimkan secara langsung ke PLC *Outseal* menggunakan protokol komunikasi serial Modbus RTU untuk mengendalikan proses pemisahan produk yang sesuai dengan warna.[1]

Proses kerja sistem konveyor ini awalnya dimulai dengan pengambilan gambar objek melalui kamera. Setelah kamera mendeteksi objek, algoritma pemrosesan citra dalam sistem akan melakukan analisis untuk mengidentifikasi dan mengklasifikasikan warna objek. Dengan menggunakan bahasa pemrograman *Python*, data yang dihasilkan dari pemrosesan gambar dapat dikirim secara langsung ke kontrol PLC (*Programmable Logic Controller*), yang bertanggung jawab untuk mengontrol aliran konveyor.[1]

Dengan menggabungkan teknologi kamera dan PLC *Outseal* melalui Modbus RTU, sistem *smart sorting* dapat mengirimkan data hasil pengolahan citra secara langsung ke PLC *Outseal*. Data tersebut dapat digunakan oleh PLC untuk

mengendalikan proses pengolahan produk secara otomatis. Dengan demikian, teknologi ini dapat membantu meningkatkan efisiensi dan produktivitas dalam proses pengolahan produk pada industri.[2]

1.2. Rumusan Masalah

Berdasarkan uraian di atas, permasalahan yang bisa diangkat pada Tugas Akhir ini adalah :

1. Bagaimana teknologi pemrosesan citra menggunakan pemrograman *Python* pada sistem *Camera Based Smart Sorting on Production Line* dapat menentukan warna dan bentuk objek yang sedang diolah?
2. Bagaimana teknologi Modbus RTU (*Remote Terminal Unit*) dapat mengirim hasil pemrosesan citra dari kamera ke PLC (*Programmable Logic Controller*) *Outseal* pada sistem *Camera Based Smart Sorting on Production Line*?
3. Implementasi *Camera Based Smart Sorting on Production Line* pada Pemrograman PLC *Outseal* untuk Pemisahan Produk Berdasarkan Warna dan bentuk.

1.3. Tujuan

Tujuan pada Tugas Akhir ini adalah:

1. Mampu mendeteksi objek berdasarkan warna dan bentuk.
2. Komputer mampu berkomunikasi dengan PLC.
3. Program mampu mengakses data pada PC melalui Modbus RTU (*Remote Terminal Unit*) untuk memisahkan objek berdasarkan bentuk bulat, kotak, dan bintang.

1.4. Manfaat

Adapun manfaat dari Tugas Akhir ini adalah mempermudah proses pemilahan produk pada area simulasi *line* produksi.

1.5. Batasan

Batasan masalah dalam Tugas Akhir ini adalah:

1. Alat hanya mampu mendeteksi benda yang berbentuk bulat, kotak, dan bintang dengan 6 variasi warna yaitu merah, merah muda, biru, biru muda, putih, dan silver.
2. Objek hanya bisa terdeteksi dari tampak atas, jika yang terdeteksi tampilan samping maka tidak akan terdeteksi oleh kamera.
3. Alat hanya mampu medeteksi objek yang memiliki diameter 40 mm dan tinggi 25 mm.
4. Sistem deteksi hanya mampu mendeteksi satu objek dalam satu *frame* kamera.

5. Ketinggian kamera pada sistem deteksi hanya mampu mendeteksi objek maksimal pada ketinggian 185 mm.
6. PC hanya mampu berkomunikasi ke PLC *Outseal* menggunakan modbus RTU (*Remote Terminal Unit*) RS485.
7. Sistem hanya dapat mengolah data jika *slave* memiliki spesifikasi data *register*.

1.6. *Work Breakdown Structure* (Opsional)

Tabel 1. *Work Breakdown Structure*

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Aditya Muhammad Kinandhika	1. Akuisisi data kamera 2. Mendeteksi objek berdasarkan warna dan bentuk
2	Putri Lidiya	1. GUI (*Graphical User Interface*) 2. Komunikasi Modbus
3	Kurniawan Teguh Setia Fatoni	1. Desain mekanikal 2. Desain eletrikal 3. Sistem Kendali *Servo* dan *Motor Stepper*

Bab 2. Tinjauan Pustaka

2.1. Penelitian Terkait

Penelitian yang dilakukan oleh Marlindia Ike Sari dkk tahun 2022 yang berjudul *THE USE OF IMAGE PROCESSING AND SENSOR IN TOMATO SORTING MACHINE BY COLOR, SIZE, AND WEIGHT*", menghasilkan proses otomatisasi dalam penyortiran tomat berdasarkan warna, ukuran, dan berat. Sistem ini akan diimplementasikan ke dalam prototipe sistem penyortiran dengan *webcam,* Arduino, *conveyor* dan *motor*.[5]

Penelitian yang dilakukan oleh Andy Suryowinoto dan Affan Zihar Wirandi tahun 2021 yang berjudul *PENGEMBANGAN SISTEM PEMILAH DAN PENGELOMPOKAN PENGHITUNG OBJEK PRODUKSI PADA KONVEYOR BERBASIS KAMERA DENGAN METODE RGB THRESHOLD,* menghasilkan pengembangan sistem berbasis pengolahan citra yang mengurutkan benda-benda produksi pada konveyor serta menghitung jumlah benda dengan menggunakan modul kemera pixy sebagai sensor utama untuk proses penyortiran dan perhitungan, penempatan barang di wadah sesuai warna barang yang dihubungkan dengan sistem *board microcontroller* Atmega328P, sehingga penelitian lebih lanjut dapat mengeksplorasi lebih bagaimana sistem berbasis pengolahan citra dalam proses penyortiran.[6]

Penelitian yang dilakukan oleh Adjhi Aprizaldi Dalimunthe dkk tahun 2019 yang berjudul *PROTOTIPE ROBOT LENGAN 3 DEGREE OF FREEDOM SEBAGAI ALAT SORTING BARANG BERDASARKAN WARNA BARANG BERBASIS INTERNET OF THINGS*, menghasilkan sistem prototipe lengan Robot 3 DOF untuk menyortir benda berdasarkan warna yang dikembangkan untuk memudahkan pekerjaan manusia dalam memisahkan benda berdasarkan warnanya. Hasil warna yang berhasil terdeteksi adalah benda berwarna, kuning, biru, dan hijau kemudian hasil akan dimasukkan kedalam kotak. Prototipe robot juga dilengkapi konveyor yang dijalankan menggunakan motor DC untuk mengantarkan benda ke titik penjemputan.[7]

Penelitian yang dilakukan oleh Nurhikma Arifin tahun 2021 yang berjudul *PROTOTYPE PEMILAH BUAH STROBERI OTOMATIS MENGGUNAKAN KAMERA BERBASIS ARDUINO UNO*. Menunjukkan pengembangan untuk membuat prototipe pemilah buah stroberi otomatis dengan menggunakan kamera berbasis Arduino Uno. Protoptipe ini menggunakan *conveyor belt, box,* LED *strip,* Arduino Uno, *Motor Servo* SG90 9G TowePro, dan kamera Logitech C920. Data stroberi diambil oleh kamera dan diolah untuk klasifikasi menggunakan algoritma SVM (*Support Vector Machine*). Hasil klasifikasi digunakan untuk memilah stroberi menjadi 2 kategori, memenuhi standar dan tidak memenuhi standar. Prototipe ini mampu memilah stroberi dengan akurasi rata-rata 95%.[8]

Berikut ini adalah perbedaan dari penelitian terdahulu yang telah dilakukan:

Tabel 2. Perbandingan Penelitian Terkait

No.	Peneliti, Judul, Tahun	Penulis	Tahun	Hasil Penelitian	Perbedaan	Kekurangan dan Kelebihan
1.	*The Use of Image Processing and Sensor in Tomato Sorting Machine by Color, Size, and Weight*	Marlindia Ike Sari, Rizal Fajar, Tedi Gunawan, Rini Handayani	2022	Pengurutan tomat berdasarkan warna dan berat berhasil digunakan dalam penelitian ini. Sistem pengurutan secara keseluruhan dianggap berhasil dan dapat dikembangkan lebih lanjut untuk penggunaan komersial. Sensor berat dan deteksi warna masing-masing memiliki akurasi 95%, sedangkan pengukuran dimensi memiliki akurasi hanya 5%.	Pada penelitian ini melakukan penentuan warna dan berat pada tomat tetapi pada tomat tidak adanya perubahan bentuk untuk dilakukan penyortiran	Kekurangan: pada penelitian terkait tidak melakukan penyortingan dalam segi bentuk, karna pada umumnya pada bentuk tomat tidak selalu bulat. kelebihan: pada sistem penelitian terkait berhasil melalukan penyortiran berdasarkan warna dan berat dengan akurasi yang tinggi yaitu 100% warna, dan 95% untuk pengukuran berat.
2.	*Pengembangan Sistem Pemilah*	Andy Suryowinoto ,	2021	Sistem konveyor berbasis kamera yang	Pada peneliatin ini	Kekurangan:

	dan Pengelompokan Penghitung Objek Produksi pada Konveyor Berbasis Kamera dengan Metode RGB Threshold	Affan Zihar Wirandi		menggunakan metode RGB *Treshold* berhasil memilih dan menghitung objek produksi. Sistem pemilihan warna kemasan sampo yang menggunakan modul kamera dan *Motor Servo* juga diuji dengan persentase keberhasilan 96% dan waktu respon rata-rata 1,72 detik. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat mendeteksi, memilah, dan menghitung barang produksi serta kemasan sampo yang tepat warnanya.	menggunakan metode RGB *Threshold* yang hanya menentukan rentang nilai piksel terhadap masing masing warna, jadi penelitian ini hanya bisa memilah berdasarkan warna saja dan tidak bisa melakukan pemilahan berdasarkan bentuk.	pada penelitian terkait respon kamera terhadap objek benda waktu merespon yaitu 1,72 detik dalam artian respon terlalu lama. Kelebihan: pembacaan deteksi objek benda memiliki angka keberhasilan yang tinggi yaitu 96%.
3.	*PROTOTIPE ROBOT LENGAN 3 DEGREE OF FREEDOM*	Adjhi Aprizaldi Dalimunthe, Naufal	2019	Prototipe Robot Lengan 3 *Degree Of Freedom* yang dapat mengangkat barang berdasarkan	Pada peneliatian ini menggunakan lengan robot	Kekurangan: dengan menggunakan lengan robot mempengaruhi proses

	SEBAGAI ALAT SORTING BARANG BERDASARKAN WARNA BARANG BERBASIS INTERNET OF THINGS	Ariyanto Adli, Taryudi		warna dengan menggunakan IoT. Pengujian dilakukan untuk IR Sensor dan Pi Kamera, dan robot dapat mengangkat barang dengan kecepatan rata-rata 7,1 detik.	untuk memisahkan barang berdasarkan warna tanpa adanya pemisahan barang berdasarkan dengan bentuk	pemilahan dengan waktu 7,1 detik/lambat. Kelebihan: pada penyortingan ini, menggunakan pengimplementasiian internet of things untuk memantau posisi dan kemajuan tempat pengambilan objek mengguanakn aplikasi blynk pada smartphone.
4.	*Prototype Pemilah Buah Stroberi Otomatis menggunakan Kamera berbasis Arduino Uno*	Nurhikma Arifin, Dian Megah Sari, Amalia Chairy	2021	Prototipe pemilah stroberi otomatis yang dibuat dengan kamera berbasis Arduino Uno memiliki akurasi rata-rata 95% dalam pemilihan stoberi, kesalahan terutama terjadi pada buah yang matang dan busuk. Metode ini dapat sangat membantu dalam otomatisasi industri	Pada penelitian ini menggunakan mikrokontoller arduino, dalam dunia otomatisasi industri menggunakan PLC	Kekurangan: tidak adanya media penempatan saat benda terdeteksi. Sehingga kemungkinan fitur warna buah, posisi dan kondisi daun stroberi bisa menimbulkan kesalahan. Kelebihan: penelitian ini menghasilkan akurasi 95%.

2.2. Dasar Teori

A. YoloV3 (*You Only Look Once version* 3)

YoloV3 (*You Only Look Once version* 3) merupakan salah satu model deteksi objek dalam bidang penglihatan komputer. Metode deteksi objek ini memungkinkan untuk mendeteksi dan megklasifikasikan berbagai objek gambar dengan cepat dan akurat. Dasar dari YoloV3 adalah konsep deteksi objek yang menggunakan *deep learning*, khususnya menggunakan CNN (*Convolution Neural Network*).[9]

YoloV3 dapat membagi gambar kedalam *grid* yang terdiri dari sel-sel kecil. Setiap sel dalam *grid* akan menjadi kandidat untuk mendeteksi objek. Kemudian untuk setiap sel Yolov3 akan melakukan prediksi *bounding box* yang menandai objek yang ada di dalam gambar. *Bounding box* ini memiliki beberapa parameter seperti koordinat, ukuran, dan skor kepercayaan (*Confidence Score*) yang menunjukkan seberapa yakin model dalam mendeteksi objek tersebut[10].

Keunggulan pada Yolov3 memiliki kemampuan dalam mendektesi objek secara singkat. Hal ini dapat mengurangi jumlah operasi yang diperlukan dalam proses deteksi objek, sehingga YoloV3 dapat memberikan kerja yang lebih cepat dalam mengakurasi deteksi gambar. YoloV3 juga dapat mendeteksi berbagai objek benda yang berbeda dalam satu gambar.

Dalam konteks *smart sorting* berbasis kamera, sistem kerja YoloV3 ini akan melakukan pemisahan gambar yang akan dibagi menjadi *grid*, melakukan ekstraksi fitur sehingga *grid* diolah oleh CNN (*Convolution Neural Network*) untuk menghasilkan fitur-fitur yang mewakili objek, melakukan prediksi objek dengan menghitung kemungkinan dan *bounding box* untuk setiap *grid*, serta menggunakan teknik *non-maximum suppression* untuk menghindari deteksi ganda agar kotak pembatas terpilih yang paling relevan.

Dengan demikian YoloV3 sangat di butuhkan pada *system smart sorting* karena YoloV3 dapat mengenali dan mengelompokkan objek secara *real-time* dengan cepat dan efisien, serta memastikan proses pengurutan berjalan dengan baik.

B. *Confusion Matrix*

Confusion matrix adalah pengukuran kinerja untuk masalah klasifikasi pembelajaran mesin di mana keluarannya bisa berupa dua kelas atau lebih, *confusion matrix* adalah rumus yang digunakan dalam melakukan pengujian penelitian ini karena dapat membantu untuk mencari tingkat akurasi dan presisi. Pada penelitian sebelumnya teknik ini belum dapat digunakan karena objek benda yang dideteksi tidak banyak, hanya 3 warna (merah, biru, putih) dan hanya

berbentuk bulat.[11] Dalam *confusion matrix* terdapat empat istilah yang digunakan untuk mewakili hasil dari proses klasifikasi, yaitu:

1) *True Positive* (TP):
Ini adalah jumlah data positif yang diklasifikasikan dengan benar oleh model sebagai positif. Dengan kata lain, model secara tepat mengidentifikasi adanya suatu peristiwa atau kondisi yang benar-benar positif.

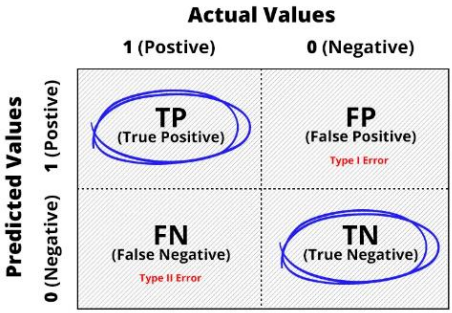
2) *True Negative* (TN):
Ini adalah jumlah data negatif yang diklasifikasikan dengan benar oleh model sebagai negatif. Dengan kata lain, model dengan benar mengenali bahwa tidak ada kejadian atau kondisi negatif.

3) *False Positive* (FP):
Ini adalah jumlah data negatif yang salah diklasifikasikan oleh model sebagai positif. Dengan kata lain, model tersebut salah mengidentifikasi adanya kejadian atau kondisi negatif sebagai kejadian atau kondisi positif.

4) *False Negative* (FN):
Ini adalah jumlah data positif yang salah diklasifikasikan oleh model sebagai negatif. Dengan kata lain, model salah mengidentifikasi peristiwa positif atau tidak adanya kondisi sebagai negatif.

Dalam melakukan pengujian terdapat 2 hal yang ingin dicari pada pengujian *Camera Based Smart Sorting on Production Line* ini, yaitu akurasi dan presisi.

(1) Akurasi
Akurasi adalah perbandingan antara jumlah prediksi yang benar (baik positif maupun negatif) dengan total jumlah data. Dengan kata lain, akurasi mencerminkan sejauh mana prediksi model mendekati nilai sebenarnya. Nilai akurasi dapat dihitung menggunakan persamaan.[11]



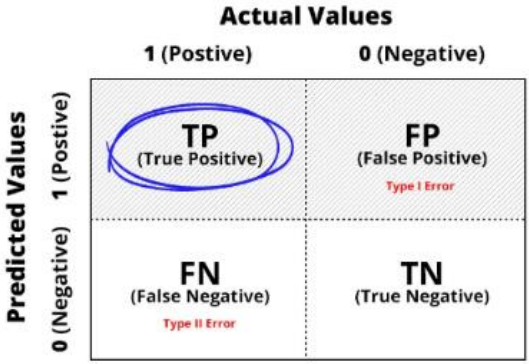
Gambar 1. *Confusion matrix* yang menggambarkan nilai akurasi.

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

Rumus tersebut berguna untuk mencari nilai akurasi dari ketepatan kamera dalam mendeteksi objek.

(2) Presisi

Menggambarkan tingkat akurasi antara data yang diminta dan hasil prediksi yang diberikan oleh model. Jadi, akurasi adalah rasio prediksi positif yang benar terhadap total hasil prediksi positif.[11]



Gambar 2. *Confusion matrix* yang menggambarkan nilai presisi.

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

Rumus tersebut berguna untuk mencari nilai presisi dari ketepatan alat dalam memilah objek.

C. *Driver Motor Stepper*

Driver Motor Stepper berfungsi untuk menggerakkan *Motor Stepper* yang mampu berputar terus menerus dengan kontrol posisi yang tepat dan tanpa adanya sistem umpan balik[12]. Sumber tegangan yang berasal dari PLC sebelum masuk pada *Motor Stepper* tegangan tersebut akan dikendalikan oleh *Driver Motor Stepper*. Pada bagian *Driver Motor Stepper* terdapat 2 bagian yaitu *microstep driver* dan PWM/ALARM. Dimana pada *microstep driver* terbagi 2 pengaturan yang pertama *subdivision setting* digunakan untuk mengatur berapa pulsa yang dikeluarkan oleh motor sedangkan *current setting* mengatur berapa sudut derajat dari *Motor Stepper*, pengaturan dari 2 bagian tersebut dari tombol SW1, SW2, dsb. Untuk PWM/ALARM terdapat bagian yaitu *signal* dan *high voltage*. *Signal* mempunyai beberapa *pin-out* yang dihubungkan ke PLC[13].

Dalam penelitian ini *Driver Motor Stepper* digunakan untuk mengontrol *Motor Servo* untuk memilah benda dan *Motor Stepper* untuk menggerakan konveyor, pada penelitian sebelumnya belum digunakan *Driver Motor Servo* karena belum ada *Motor Stepper* atau konveyor [14].

Gambar 3. *Driver Motor Stepper*

D. Komunikasi Modbus RTU (*Remote Terminal Unit)* RS4-85

Komunikasi Modbus RTU (*Remote Terminal Unit*) RS485 digunakan sebagai komunikasi *serial* yang berfungsi mengirimkan data dari PC ke PLC (*Programmable Logic Controller*). Komunikasi Modbus RTU (*Remote Terminal Unit*) RS485 menggunakan sistem transmisi data dengan *half-duplex methode*, yang berarti data hanya dapat dikirim atau diterima pada satu waktu[15].

Protokol ini menggunakan sebuah *master* (biasanya PLC) dan satu atau lebih *slave* (biasanya sensor atau aktuator) yang terhubung melalui jalur komunikasi

RS485. Pada projek ini Modbus RTU (*Remote Terminal Unit*) RS4-85 digunakan karena lebih kompatibel untuk komunikasi dari PLC *Outseal* ke kamera dan juga ke PC, di penelitian sebelumnya menggunakan Adam *Input/Output* 6050 karena belum menggunakan PLC dan *Input/Output* nya belum mencukupi.[16]

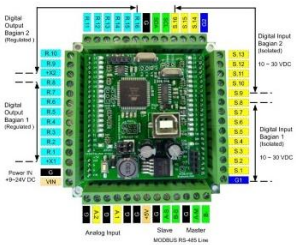
Berikut adalah program *Python* untuk menghubungkan Modbus RTU (*Remote Terminal Unit)* ke PLC:

```
import minimalmodbus
import time

baudrate = 115200
instrument = minimalmodbus.Instrument('COM5', 1, minimalmodbus.MODE_RTU, debug = True)
# port name, slave address (in decimal)

instrument.serial.baudrate = baudrate

while True:
## Read temperature (PV = ProcessValue) ##
  instrument.write_register(1, 1, 0, 6)  # Registernumber, number of decimals
  time.sleep(500)
  instrument.write_register(1, 0, 0, 6)  # Registernumber, number of decimals
  time.sleep(500)
```

Gambar 4. Komunikasi modbus RTU (*Remote Terminal Unit)* RS485 pada PLC

E. PLC *(Programmable Logic Controller) Outseal*

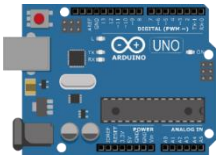
PLC (*Programmable Logic Controller*) *Outseal* merupakan pengontrol pengembangan teknologi otomasi yang berasal dari Indonesia. Teknologi ini dapat digunakan dalam sistem otomatisasi untuk mengendalikan berbagai proses. PLC *Outseal* ini bertujuan untuk menggerakkan alat *Camera Based Smart Sorting on Production Line*. PLC ini memiliki *input* dan *output*. *Input* berupa tombol *start*, tombol *stop*, tombol *emergency*, dan sensor[17]. Sedangkan, *output* berupa lampu *tower*, dan *Motor Stepper. Input/output* akan disatukan untuk menjalankan alat. Namun untuk menjalankan tersebut membutuhkan bantuan *software* yang bernama *Outseal* Studio. *Outseal* Studio ini berisi program *ladder* diagram, yang membantu *user* memudahkan pengerjaan sistem otomatisasi untuk mengendalikan alat yang akan dikerjakan[2].



Gambar 5. PLC *Outseal*

F. Arduino

Arduino merupakan perangkat keras yang dirancang untuk memudahkan pengembangan dan *protoyping* proyek-proyek elektronik. Ini terdiri dari papan sirkuit cetak berukuran kecil yang dilengkapi dengan mikrokontroler dan sejumlah *pin input/output* yang dapat digunakan untuk menghubungkan sensor, aktuator, dan komponen elektronik lainnya[18]. Untuk menggerakan *Servo* pada alat ini, ardunio berperan sebagai pengatur derajat pada *Servo* melalui program Arduino IDE[19].

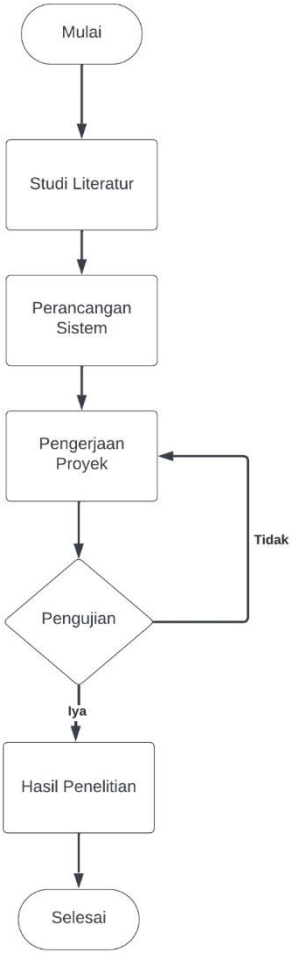


Gambar 6. Arduino

Bab 3. Metodologi Penelitian

3.1. Perancangan

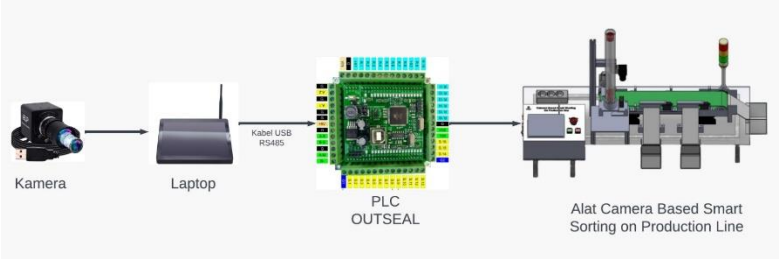
Agar proses pengerjaan menjadi lebih mudah, diperlukan metode perancangan yang terurut dan jelas, seperti yang dijelaskan di bawah ini:



Gambar 7. *Flowchart* Perancangan Alat

Pada awal saat akan mengerjakan tugas akhir studi literatur dilakukan di awal penelitian untuk memperoleh referensi dari berbagai sumber, seperti melalui jurnal-jurnal dan situs *web*, agar mendapatkan data dan teori yang berkaitan dengan sistem yang akan dirancang. Setelah itu, barulah dilakukan perancangan sistem dari proses *input* hingga mencapai *output* yang diinginkan.

Camera based smart sorting on production line mempunyai sistem yang saling terhubung satu sama lain. PLC menggunakan kabel USB RS485 yang terhubung dengan laptop dengan begitu PLC dapat diberikan perintah untuk menjalankan alat. Berikut ini Topologi dari *camera based smart sorting on production line.*

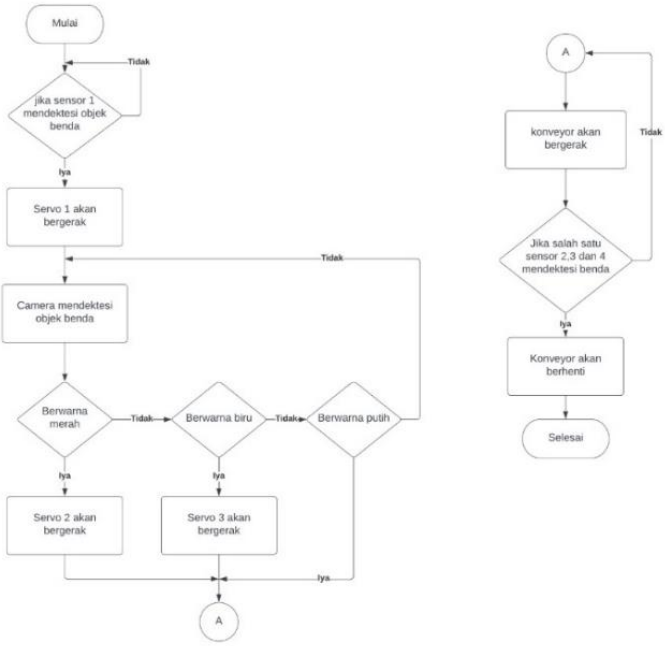


Gambar 8. Topologi Diagram Alat *Camera Based Smart Sorting on Production Line*

Setelah selesai melakukan perancangan, selanjutnya adalah pembuatan program menggunakan algoritma *Python* dengan langkah-langkah sebagai berikut:

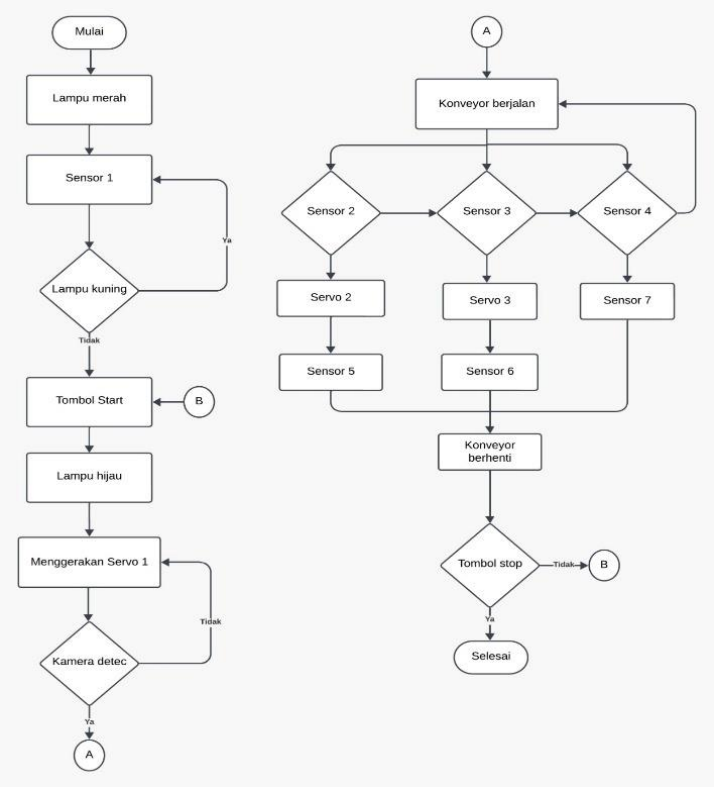
a. Membuat aplikasi yang menggunakan kamera sebagai *input*.
b. Mengambil gambar dari kamera.
c. Mengolah gambar yang diambil dengan menggunakan *OpenCV* untuk mendeteksi objek yang berada pada gambar tersebut.
d. Membuat algoritma untuk mengklasifikasikan objek yang telah terdeteksi berdasarkan fitur-fiturnya.
e. Menentukan kategori atau label untuk setiap objek berdasarkan hasil klasifikasi.
f. Mengirimkan informasi mengenai kategori tersebut ke perangkat PLC *Outseal* untuk dilakukan pemilahan.

Berikut ini adalah diagram alir sistem kerja alat dan sistem kerja program:



Gambar 9. *Flowchart* Sistem Kerja Alat

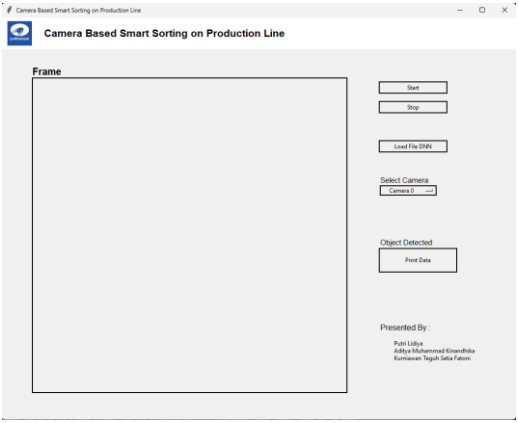
Berikut ini adalah diagram alir sistem kerja program PLC:



Gambar 10. *Flowchart* Sistem Kerja Program

A. Perancangan GUI (*Graphical User Interface*)

GUI (*Graphical User Interface*) ini nantinya dilakukan melalui program pada aplikasi *Visual Studio Code*. Pada pembuatan GUI (*Graphical User Interface*) nantinya akan memiliki memiliki ukuran 1050 x 820 piksel. Pada bagian kamera dibuat *outline* sebagai penanda posisi kamera yang memiliki ukuran 640 x 640 piksel, sedangkan pada ukuran kamera nya sendiri memiliki ukuran 640 x 480 piksel.



Gambar 11. Tampilan GUI (*Graphical User Interface*)

Gambar 11 merupakan desain awal tampilan GUI (*Graphical User Interface*) yang diusulkan pada Tugas Akhir ini. Adapun bagian-bagiannya adalah sebagai berikut.

Tabel 3. Bagian-bagian GUI (*Graphical User Interface*)

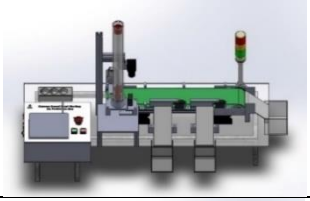
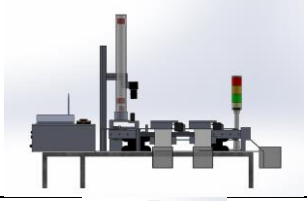
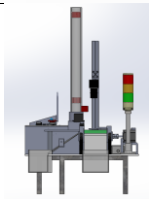
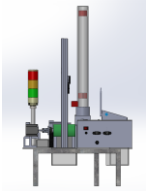
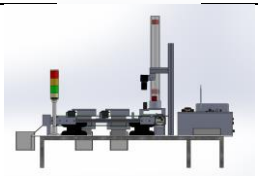
No.	Bagan	Fungsi
1.	Judul "*Camera Based Smart Sorting on Production Line*"	Membantu pembaca untuk memiliki pemahaman awal tentang topik atau isi tulisan.
2.	*Frame*/Layar kamera	Menampilkan gambar yang ditangkap oleh kamera.
3.	Tombol *Start*	Untuk mengaktifkan kamera.
4.	Tombol *Stop*	Untuk mematikan kamera.
5.	Tombol *Load File DNN*	Untuk pengambilan *file* data objek yang sudah berhasil di deteksi oleh kamera, pada program GUI (*Graphical User Interface*) akan diberi fungsi *Load File* untuk mengambil *file* data objek tersebut. Jika ada objek baru yang akan di deteksi selanjutnya juga bisa di ganti atau di tukar pada *Load File DNN* tersebut.
6.	Tombol *Select Camera*	Memberikan opsi untuk memilih kamera memungkinkan pengguna untuk menggunakan kamera yang sesuai dengan preferensi atau kebutuhan mereka.

		Misalnya, jika pengguna memiliki beberapa kamera terhubung ke komputer mereka, mereka dapat memilih kamera yang ingin mereka gunakan.
7.	*Frame 'Objected Detected'*	Hasil *print* data yang akan ditampilkan bisa dilihat pada *frame Object Detected*. Hasil yang akan terdeteksi yaitu *box_blue, box_light_blue, box_pink, box_red, box_white, box_silver, circle_blue, circle_light_blue, circle_pink, circle_red, circle_white, circle_silver, star_blue, star_ligt_blue, star_pink, star_red, star_white, star_silver.*

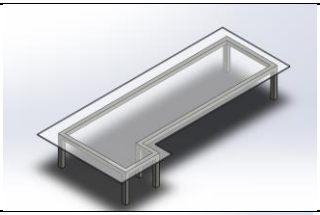
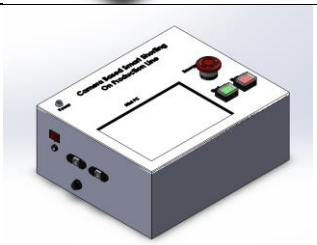
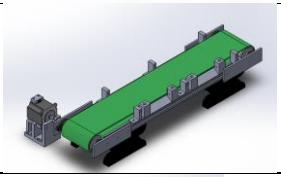
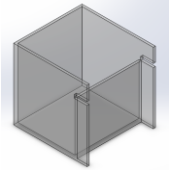
B. Bentuk Gambaran Alat Dari Semua Sisi

Berikut ini adalah bentuk gambaran alat dari berbagai sisi.

Tabel 4. Gambar alat dari berbagai sisi

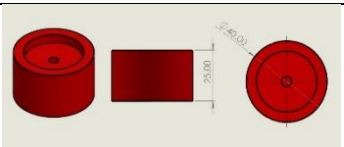
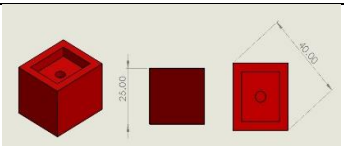
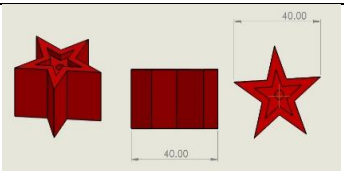
No	Nama	Gambar
1	Desain mekanikal bagian depan atas	
2	Desain mekanikal bagian depan	
3	Desain mekanikal bagian sisi kanan	
4	Desain mekanikal bagian sisi kiri	
5	Desain mekanikal bagian belakang	

Tabel 5. Bagian Mekanikal alat

NO	Bagian mekanikal	Fungsi	Gambar
1	Meja	Sebagai peletakan alat	
2	Box panel	Sebagai penempatan eletrikal	
3	Konveyor	Sebagai pemindahan objek benda	
4	Box penempatan	Sebagai tempat objek yang sudah terpilah	
5	Akrilik bulat	Sebagai masuknya objek benda	

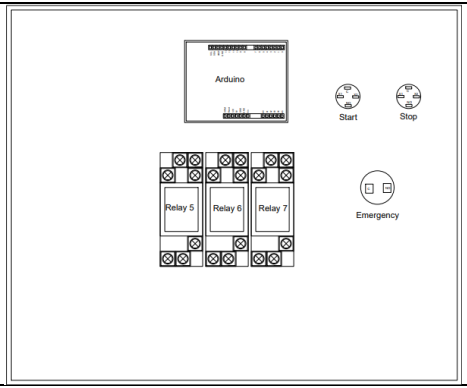
Berikut ini *drawing* objek yang dideteksi oleh *Camera Based Smart Sorting on Production Line*. Gambar 15 merupakan ukuran objek berbentuk bulat, gambar 16 ukuran objek berbentuk kotak, dan gambar 17 ukuran objek berbentuk bintang pada *Camera Based Smart Sorting on Production Line* dalam ukuran millimeter.

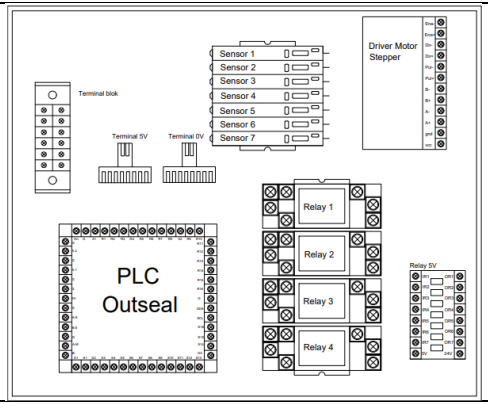
Tabel 6. Bentuk Objek deteksi

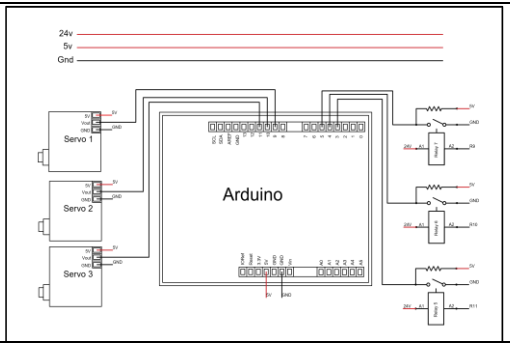
No	Bentuk	Gambar
1	Bulat	
2	Kotak	
3	Bintang	

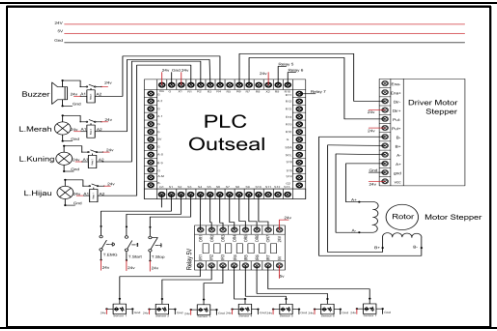
C. Perancangan Elektrikal

Tabel 7. Perancangan Elektrikal

No	Nama	Gambar
1	Penempatan komponen pada tutup panel	
Keterangan		Gambaran rangkaian perancangan elektrikal pada "*Camera Based Smart Sorting on Production Line* " Pada penempatan desain elektrikal pada tutup panel terdapat a. Arduino untuk mengontrol pergerkan *Servo.* b. 3 buah *relay* sebagai input untuk menggerkan *Servo, Servo* pada alat ini berfungsi untuk mendorong objek benda dan memilah objek benda. c. Tombol *start* untuk menjalankan alat *Camera Based Smart Sorting on Production Line.*

		d. Tombol *stop* untuk memberhentikan jalannya suatu alat. e. Tombol *emergency* untuk memberhentikan alat secara paksa dan mereset semua alat.
2	Penempatan komponen pada box panel	
Keterangan		Terdapat 4 *relay* untuk menghidupkan lampu merah sebagai penanda bahwa alat tidak berjalan, lampu kuning mengetahui bahwa objek benda belum berada pada penempatan benda, lampu hijau mengetahu bahwa alat sedang berjalan, dan *buzzer* memperingati bahwa alat sedang *eror* atau objek benda tidak berada di penempatannya. Terpasang 7 sensor untuk mendeteksi keberadaan objek benda, yaitu ; a. Sensor 1 untuk mendeteksi keberadaan benda pada selongsong benda, b. Sensor 2 mengetahui keberdaan benda pada konveyor, c. Sensor 3 mengetahui keberadaan benda lewat pada tempat pemilah 1, d. Sensor 4 mengetahui benda yang sudah terpilah ditempat pemilihan 1,

		e. Sensor 5 untuk mengetahui benda yang lewat terhadap tempat pemilah 2, f. Sensor 6 untuk mengetahui objek benda yang sudah terpilah pada tempat pemilahan 2 dan g. Sensor 7 untuk mengetahui objek benda yang sudah terpilah ke tempat pemilahan 3 Lalu terdapat *Driver Motor Stepper* untuk mengatur gerakan *Motor Stepper*. Pada modul *relay 5V* berfungsi untuk mengubah nilai tegangan negatif dari sensor menjadi tegangan positif yang akan diberikan ke pin input PLC *Outseal*. Sedangkan pada terminal blok berfungsi untuk memberikan tempat *24V* dan *gnd*.
3	Tampilan eletrikal pada Arduino	
Keterangan		Pada *wiring* diagram membutuhkan tegangan 5V untuk mengaktifkan Arduino. Dimana pada Arduino memiliki 14 *pin input/output* dan memiliki 6 *pin analog*. Pada kontrol *Motor Servo pin* 3, 4 dan 5 sebagai *input* dari anak kontak *relay* yang berfungsi untuk mengaktifkan salah satu *pin* 9, 10 dan 11 sebagai *output*. Dimana *pin* 9, 10 dan 11 berfungsi untuk mengaktifkan *Servo*. Pada tegangan 24V disini berfungsi untuk mengaktifkan A1 dari *relay*, pada A2 *relay* akan diaktifkan oleh *output* dari PLC *Outseal*.

<table>
<tr><td>4</td><td>Tampilan elektrikal pada PLC Outseal</td><td> 
</td></tr>
<tr><td colspan="2">Keterangan</td><td>Pada wiring diagram PLC Outseal terdapat 16 pin input dan 16 pin output untuk mengaktifkan PLC Outseal.
a. Untuk mengaktifkan PLC Outseal akan diberikan tegangan 24V pada pin Vin,
b. Untuk mengaktifkan input PLC pada G1 dan G2 diberikan Ground,
c. Output PLC akan aktif jika X1 dan X2 diberikan tegangan 24V,
d. Pada pin input S1 terhubung dengan tombol emergency.
e. S2 terhubung dengan tombol start,
f. S3 terhubung dengan tombol stop,
g. S4-S10 terhubung dengan output relay 5V (OR1-OR7).
Pada PLC Outseal pin output yang digunakan yaitu R1, R2, R3, R4, R6, R7, R9, R10, dan R11. Dimana :
a. R1 terhubung dengan A2 pada relay 1
b. R2 terhubung dengan A2 pada relay 2
c. R3 terhubung dengan A2 pada relay 3
d. R4 terhubung dengan A2 pada relay 4</td></tr>
</table>

	e. R6 terhubung dengan Dir- pada *Driver Motor Stepper* f. R7 terhubung dengan Pul- pada *Driver Motor Stepper* g. R9 terhubung dengan A2 pada *relay 5* h. R10 terhubung dengan A2 pada *relay 6* i. R11 terhubung dengan A2 pada *relay 7* Pada Sensor 1-7 akan terhubung dengan *input relay 5V* (IR1-IR7). Pada anak kontak *relay 1 NO* terhubung dengan lampu hijau, pada anak kontak *relay 2 NO* terhubung dengan lampu kuning, anak kontak *relay 3 NO* terhubung dengan lampu merah, sedangkan anak kontak *relay 4 NO* terhubung dengan *buzzer*. Untuk mengaktifkan *Driver Motor Stepper* memerlukan tegangan 24V pada *pin VCC*. Pada *Motor Stepper* terdapat kabel A+, A- B+, dan B- yang akan terhubung ke *Driver Motor Stepper*.

3.2. Alat dan Bahan (Opsional)

Tabel di bawah ini merupakan estimasi biaya yang diperlukan dalam pembuatan alat "*Camera Based Smart Sorting on Production Line* " diantaranya:

Tabel 8. Estimasi biaya

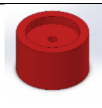
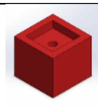
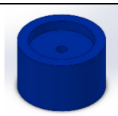
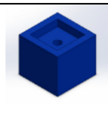
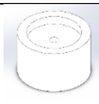
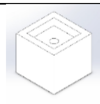
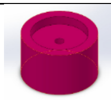
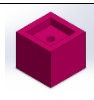
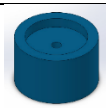
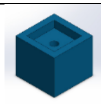
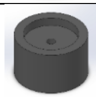
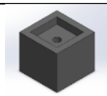
No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1	Camera ELP 8 Megapixel	1.700.000	1	1.700.000	Disediakan Kampus
2	Mini PC	2.500.000	1	2.500.000	Disediakan Kampus
3	PLC *Outseal* mega V1.1	800.000	1	800.000	Disediakan Kampus
4	Lampu Tower 24volt	600.000	1	600.000	
5	Adaptor 24v	100.000	1	100.00	
6	Motor stepper	400.000	1	400.000	
7	Driver Motor Stepper	100.000	1	100.000	Disediakan Kampus
8	Modul relay 24v	30.000	2	60.000	
9	Modul PCA9685	100.000	1	100.000	
10	Modul modbus RTU	100.000	1	100.000	
11	Sensor proximity	120.000	4	480.000	
12	Servo MG995	50.000	2	100.000	
13	Aluminum profil 40x40 (50cm)	100.000	1	100.000	
14	Box Akrilik	250.000	1	250.000	
15	Conveyor	500.000	1	490.000	
16	Servo	50.000	3	150.000	
17	Akrilik	150.000	1	150.000	
18	Kabel jumper	10.000	10	100.000	
19	Red button	30.000	1	30.000	
20	Green button	30.000	1	30.000	
21	Emergency button	50.000	1	50.000	
21	Sensor Optik	850.000	7	5.950.000	
22	Filamen	280.000	3	840.000	
23	Relay	60.000	4	240.000	
	Total			15.420.000	

3.3. Pengujian

A. Pengujian pada objek benda

Pada pengujian ini akan didapatkan hasil *confusion matrix* dari setiap percobaan pada objek benda, pengujian *confusion matrix* berguna untuk mengetahui nilai akurasi dan presisi pada setiap objek benda. Berikut ini adalah setiap objek benda yang akan dideteksi oleh kamera.

Tabel 9. Variasi warna dan bentuk objek

Bulat	Kotak	Bintang
		
		
		
		
		
		

B. Pengujian Komunikasi Modbus RTU (Remote Terminal Unit)

Pada Pengujian Komunikasi Mobus RTU (*Remote Terminal Unit)* menggunakan kabel USB sebagai penghubung pada Modbus untuk bisa berkomunikasi dari PC ke PLC *Outseal*. Pengujian program komunikasi dilakukan menggunakan *software Visual Studio Code*. Pada program diberikan settingan penghubung antara program dengan PLC *Outseal*. Untuk pengaturan tersebut di bagi mejadi dua *master* dan *slave*. *Master* (PC) mengirim perintah ke *Slave* (PLC) untuk

memisahkan data objek. Sedangkan *Slave* (PLC) menerima dan menjalankan perintah yang diberikan. Jika kedua nya saling mengirim dan menerima maka pengujian komunikasi modbus RTU (*Remote Terminal Unit)* dikatakan berhasil.

C. Pengujian Kendali *Motor Stepper* dan *Motor Servo*

Pengujian ini akan bisa mengendalikan kecepatan *Motor Stepper* melalui program *Outseal* dengan mengatur seberapa besar frekuensi yang digunakan untuk menjalankan *Motor Stepper*. Pada pengujian *Motor Servo* dapat mengatur seberapa besar derajat yang digunakan untuk menggerakan *Servo* melalui program Arduino.

Bab 4. Hasil dan Pembahasan

Data hasil penelitian merupakan hasil dari pengujian yang telah dilakukan terhadap perancangan yang telah dibuat. Pengujian terdiri dari pengujian akurasi dan presisi pada benda objek, akuisisi data gambar dari kamera, pendeteksian objek berdasarkan warna dan bentuk, *GUI (Graphical User Interface)*, program Komunikasi modbus, desain mekanikal, perancangan elektrikal pada alat, serta kendali *Motor Stepper* dan *Motor Servo.*

4.1 Hasil pengujian pada benda

(1) Pengambilan data objek benda

Hasil Pengujian pada objek benda telah dilakukan pengujian pada alat dengan berdasarkan hasil data percobaan. Setiap objek dilakukan 50 kali percobaan. Sehingga mendapatkan hasil dari *confusion matrix* di setiap percobaan pada objek.

Tabel 10. *Confusion Matrix* pada objek benda

x		Predicted																	
		Box Blue	Box Light Blue	Box Pink	Box Red	Box Silver	Box White	Circle Blue	Circle Light Blue	Circle Pink	Circle Red	Circle Silver	Circle White	Star Blue	Star Light Blue	Star Pink	Star Red	Star Silver	Star White
Actual	Box Blue	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Box Light Blue	3	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Box Pink	0	0	47	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
	Box Red	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Box Silver	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
	Box White	0	0	0	0	0	46	0	0	0	0	0	4	0	0	0	0	0	0
	Circle Blue	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
	Circle Light Blue	0	0	0	0	0	0	3	47	0	0	0	0	0	0	0	0	0	0
	Circle Pink	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0
	Circle Red	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0
	Circle Silver	0	0	0	0	0	0	0	0	5	0	42	3	0	0	0	0	0	0
	Circle White	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
	Star Blue	0	0	0	0	0	0	1	0	0	0	0	0	49	0	0	0	0	0
	Star Light Blue	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0
	Star Pink	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0
	Star Red	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	49	0	0
	Star Silver	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	47	2
	Star White	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50

Pada tabel diatas dapat di jelaskan sebagai berikut :

a. *Box blue* tidak ada kesalahan setiap percobaan
b. *Box light blue* terjadi kesalahan dalam percobaan dengan membaca *box blue* 3 kali
c. *Box pink* terjadi kesalahan dalam percobaan dengan membaca *circle pink* 3 kali
d. *Box red* tidak ada kesalahan setiap percobaan
e. *Box silver* tidak ada kesalahan setiap percobaan
f. *Box white* terjadi kesalahan dalam percobaan dengan membaca *circle white* 4 kali
g. *Circle blue* tidak ada kesalahan setiap percobaan
h. *Circle light blue* terjadi kesalahan dalam percobaan dengan membaca *circle blue* 3 kali
i. *Circle pink* tidak ada kesalahan setiap percobaan
j. Pada *circle red* tidak ada kesalahan setiap percobaan
k. *Circle silver* terjadi kesalahan dalam percobaan dengan membaca *5 circle pink* dan 3 *circle white*
l. *Circle white* tidak ada kesalahan setiap percobaan
m. *Star blue* terjadi kesalahan dalam percobaan dengam membaca *circle blue* 1 kali
n. *Star light blue* tidak ada kesalahan setiap percobaan
o. *Star pink* tidak ada kesalahan setiap percobaan
p. *Star red* terjadi kesalahan dalam percobaan dengan membaca *star pink* 1 kali
q. *Star silver* terjadi kesalahan dalam percobaan dengan membaca *star pink* 2 kali dan *star white* 1 kali
r. *Star white* tidak adakesalahan setiap percobaan

Pada hasil *confusion matrix* dapat mencari nilai akurasi dan presisi pada setiap objek, pada perhitungan dapat menggunkan rumus pada halaman 6 dan 7, berikut perhitungan untuk pengambilan nilai akurasi dan presisi :

Tabel 11. Hasil akurasi dan presisi pada setiap objek benda

NO	Nama	Hasil perhitungan akurasi	Hasil Perhitungan presisi
1	*Box Blue*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(3)} = 0.99 =$ 99%	Presisi $= \frac{TP}{(TP+FP)} = \frac{50}{50+0} = \frac{50}{50} = 1 =$ 100%
2	*Box Light Blue*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(3)} = 0.99 =$ 99%	Presisi $= \frac{TP}{(TP+FP)} = \frac{47}{47+(3)} = 0.94 = 94\%$
3	*Box Pink*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(3)} = 0.99 =$ 99%	Presisi $= \frac{TP}{(TP+FP)} = \frac{47}{47+(3)} = 0.94 = 94\%$
4	*Box Red*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi $= \frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
5	*Box Silver*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi $= \frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
6	*Box White*	Akurasi $= \frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi $= \frac{TP}{(TP+FP)} = \frac{46}{46+(4)} = 0.92 = 92\%$

7	*Circle Blue*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(4)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
8	*Circle Light Blue*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(3)} = 0.99 = 99\%$	Presisi = $\frac{TP}{(TP+FP)} = \frac{47}{47+(3)} = 0.94 = 94\%$
9	*Circle Pink*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(8)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
10	*Circle Red*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
11	*Circle Silver*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(8)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{42}{42+(8)} = 0.84 = 84\%$
12	*Circle White*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
13	*Star Blue*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(1)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{49}{49+(1)} = 0.98 = 98\%$
14	*Star Light Blue*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(0)} = 1 = 100\%$	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$

15	Star Pink	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(2)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$
16	*Star Red*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(1)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{49}{49+(1)} = 0.98 = 98\%$
17	*Star Silver*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(3)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{47}{47+(3)} = 0.94 = 94\%$
18	*Star White*	Akurasi = $\frac{(TP+TN)}{TP+TN+FP+FN} = \frac{874}{874+(2)} = 0.99 =$ 99%	Presisi = $\frac{TP}{(TP+FP)} = \frac{50}{50+(0)} = 1 = 100\%$

Tabel dibawah ini adalah hasil perhitungan nilai presisi dan akurasi yang didapat pada setiap objek benda, dimana nilai presisi dan akurasi terbilang baik dikarenakan tidak adanya nilai akurasi dan presisi yang dibawah 80%.

Tabel 12. Pengujian perhitungan benda terhadap presisi dan akurasi

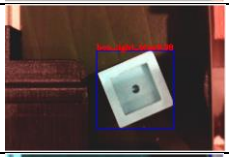
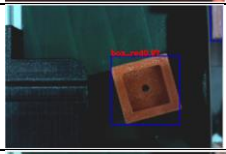
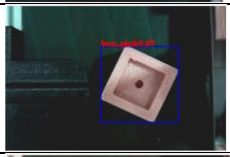
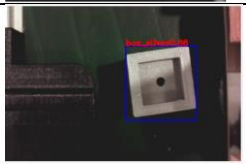
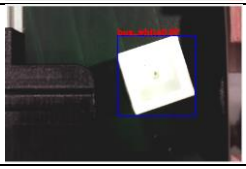
No	Benda	Akurasi(%)	Presisi(%)
1	Box Blue	99	100
2	Box Light Blue	99	94
3	Box Pink	99	94
4	Box Red	100	100
5	Box Silver	100	100
6	Box White	100	92
7	Cricle Blue	99	100
8	Cricle Light Blue	99	94
9	Cricle Pink	99	100
10	Cricle Red	100	100
11	Cricle Silver	99	84
12	Cricle White	100	100
13	Star Blue	99	98
14	Star Light Blue	100	100
15	Star Pink	99	100
16	Star Red	99	98
17	Star Silver	99	94
18	Star White	99	100

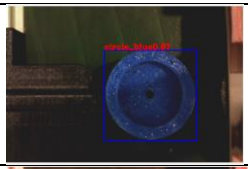
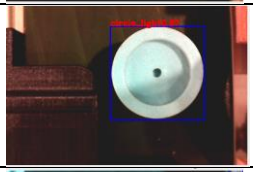
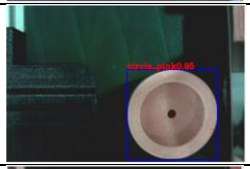
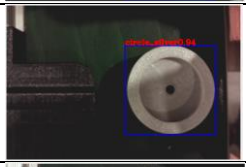
(2) Pengambilan objek benda pada kamera

Hasil ini merupakan hasil pengujian deteksi objek pada kamera yang telah dilakukan. Agar kamera dapat mendeteksi sebuah objek benda, dilakukan pengambilan *data train* sebanyak 150 kali menggunakan aplikasi *Roboflow*. *Roboflow* dapat membantu dalam pengolahan gambar dan pelabelan dataset. Pada pelabelan dataset ini menggunakan fitur bernama *Anotation Dataset*. Anotasi *Dataset* ini dapat mengunggah *dataset* gambar dan melakukan dengan mudah. Dapat menandai objek, menggambar kotak batas, dan memberikan label pada gambar.

Dengan melakukan pelabelan *dataset* sebanyak 150 kali pada percobaan ini, kamera mampu medeteksi objek dengan jelas pada saat benda sedang di area. Berikut tabel hasil kamera yang mampu mendeteksi sebuah objek benda :

Tabel 13. Hasil deteksi objek

No.	Nama Objek	Hasil detected objek	Kondisi
1.	*Box_blue*		OK
2.	*Box_light_blue*		OK
3.	*Box_red*		OK
4.	*Box_pink*		OK
5.	*Box_silver*		OK
6.	*Box_white*		OK

7.	*Circle_blue*		OK
8.	*Circle_light_blue*		OK
9.	*Circle_red*		OK
10.	*Circle_pink*		OK
11.	*Cirlce_silver*		OK
12.	*Circle_white*		OK
13.	*Star_blue*		OK

14.	*Star_light_blue*		OK
15.	*Star_red*		OK
16.	*Star_pink*		OK
17.	*Star_silver*		OK
18.	*Star_White*		OK

Dari hasil tabel pengujian diatas bahwa setiap objek benda yang di deteksi kamera sesuai dengan objek benda, dengan begitu deteksi pada objek benda terbilang berhasil.

4.2 Hasil Komunikasi Data Modbus dari PC ke PLC (*Programmable Logic Controller*)

Hasil Komunikasi Modbus RTU (*Remote Terminal Unit)* ini merupakan hasil pengujian komunikasi modbus yang telah berhasil terhubung antara PC ke PLC (*Programmable Logic Controller*) melalui program sehingga alat bisa mengirim dan menerima perintah untuk memisahkan data objek benda yang telah terdeteksi oleh data kamera.

Tabel 14. Hasil Pembacaan input dari Kamera GUI (*Graphical User Interface*) ke PLC (*Programmable Logic Controller*)

Object	Monitoring GUI	Alamat input PLC Integer	Status input PLC Integer	Status
Box_Blue	Detect	i.1	On	Succes
Box_Light_Blue	Detect	i.2	On	Succes
Box_Pink	Detect	i.3	On	Succes
Box_Red	Detect	i.4	On	Succes
Box_Silver	Detect	i.5	On	Succes
Box_White	Detect	i.6	On	Succes
Circle_Blue	Detect	i.7	On	Succes
Circle_Light_Blue	Detect	i.8	On	Succes
Circle_Pink	Detect	i.9	On	Succes
Circle_Red	Detect	i.10	On	Succes
Circle_Silver	Detect	i.11	On	Succes
Circle_White	Detect	i.12	On	Succes
Star_Blue	Detect	i.13	On	Succes
Star_Light	Detect	i.14	On	Succes
Star_Pink	Detect	i.15	On	Succes
Star_Red	Detect	i.16	On	Succes
Star_Silver	Detect	i.17	On	Succes
Star_White	Detect	i.18	On	Succes

Dari tabel yang telah dilampirkan, dapat disimpulkan bahwa setiap objek yang terdeteksi oleh kamera dapat di *monitoring* oleh GUI (*Graphical User Interface*) secara *real time* dengan begitu data yang dikirim akan mengaktifkan data *input* dari PLC (*Programmable Logic Controller*) dengan kondisi *On*. Maka semua pengiriman data berhasil atau dikatakan *success.*

Tabel 15. Hasil komunikasi data modbus

NO	Trial	DATA FRAME MODBUS RTU					Alamat input integer pada PLC	Object Benda
		Slave ID	Function code	Data		CRC		
				Register Address	Preset Data			
1	a	01	06	00 01	00 01	19 CA	i.1	Box_Blue
2	b	01	06	00 02	00 01	E9 CA	i.2	Box_Light_Blue
3	c	01	06	00 03	00 01	B8 0A	i.3	Box_Pink
4	d	01	06	00 04	00 01	09 CB	i.4	Box_Red
5	e	01	06	00 05	00 01	58 0B	i.5	Box_Silver
6	f	01	06	00 06	00 01	A8 0B	i.6	Box_White
7	g	01	06	00 07	00 01	F9 CB	i.7	Circle_Blue
8	h	01	06	00 08	00 01	C9 C8	i.8	Circle_Light_Blue
9	i	01	06	00 09	00 01	98 08	i.9	Circle_Pink
10	j	01	06	00 0A	00 01	68 08	i.10	Circle_Red
11	k	01	06	00 0B	00 01	39 C8	i.11	Circle_Silver
12	l	01	06	00 0C	00 01	88 09	i.12	Circle_White
13	m	01	06	00 0D	00 01	D9 C9	i.13	Star_Blue
14	n	01	06	00 0E	00 01	29 C9	i.14	Star_Light
15	o	01	06	00 0F	00 01	78 09	i.15	Star_Pink
16	p	01	06	00 10	00 01	49 CF	i.16	Star_Red
17	q	01	06	00 11	00 01	18 0F	i.17	Star_Silver
18	r	01	06	00 12	00 01	E8 0F	i.18	Star_White

Dari tabel di atas dapat disimpulkan bahwa pengujian komunikasi data modbus dapat dikirim melalui *master* (PC) ke PLC (*Programmable Logic Controller*) sebagai *slave*.

Trial diatas sebagai percobaan untuk mengetahui data yang dikirim sesuai dengan jumlah objek yang ditentukan yaitu 18 objek benda, dengan menggunakan variabel a-r. Dalam percobaan ini terdapat *data frame modbus* sebagai pemanggil data dari *master* (PC) menuju *slave* PLC (*Programmable Logic Controller*). Sehinga data kamera tersebut yang sudah disesuaikan akan membaca objek benda yang telah ditentukan oleh program.

Sebagai contoh, perhatikan pada baris no 2 Nilai " 01 06 00 02 00 01 E9 CA" yang merupakan hasil data *frame master to slave*.

Tabel 16. Salah satu percobaan *Master to Slave*

Nilai Data *Master to Slave*	Keterangan
01	Alamat *slave* PLC (*Programmable Logic Controller*) *Outseal* yang menjadi target perintah Modbus. Nilai 01

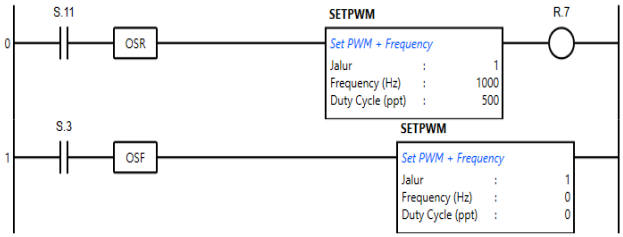
	menunjukkan bahwa perintah ini ditunjukkan ke *slave* dengan alamat 01.
06	Kode fungsi Modbus ini menunjukkan bahwa perintah ini akan mengubah nilai suatu *register* di PLC (*Programmable Logic Controller*) *Outseal* = 6 (*integer*).
00 02	Alamat pada PLC (*Programmable Logic Controller*).
00 01	Data yang mengatur nilai alamat ke parameter nilai 1 sehingga alamat aktif.
E9 CA	Ini merupakan nilai CRC (*Cyclic Redundancy Check*) untuk memastikan integritas data dalam *frame* (memastikan bahwa data informasi yang dikirim atau diterima tidak mengalami perubahan, kerusakan, atau manipulasi selama proses *transmisi*).

4.3 Hasil pengendali *Motor Stepper* dan *Servo*

1. Pengendali *Motor Stepper*

Proses pengujian pada kendali *Motor Stepper* dengan menggunakan *start-stop* untuk menggerakan motor dan menghentikan motor. Proses pengujian ini dilakukan dengan melakukan konfigurasi parameter pada instruksi SETPWM di *software Outseal Studio*. Pada software ini untuk mengatur frekuensi dan *duty-cycle.* Pada PWM (*Pulse Width Modulation*) berguna sebagai mengatur durasi sinyal *on* dan *off* dalam gelombang pulsa, dimana nilai akan mengambarkan perbandingan on terhadap panjang gelombang yang disebut dengan *duty-cycle,* frekuensi banyaknya gelombang dalam satu detik. Nilai frekuensi dapat diisi dengan 0-8Khz. Sedangkan *duty-cycle* berupa bentuk persen (%), melainkan dalam ppt (*point per thousand*) atau 1/1000.

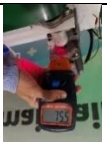
Berikut tampilan *ladder diagram* untuk pengatur SETPWM pada *Outseal* Studio.



Gambar 12. Program *Motor Stepper* dengan *software Outseal Studio*

Pengujian kendali kecepatan putaran pada *Motor Stepper* dilakukan dengan mengukur nilai RPM motor yang di hasilkan dari kendali *start-stop* dengan memberikan nilai frekuensi. Pengukuran RPM dilakukan dengan alat ukur *tachometer*, pengukuran ini bertujuan untuk mengetahui nilai RPM dari *Motor Stepper* dan kondisi pada *Motor Stepper*.

Tabel 17. Data pengukuran kecepatan pada *Motor Stepper*

No	Set Frekuensi	Rpm	Kondisi *Motor Stepper*	Gambar pengukuran
1	500	32.2	Motor berputar	
2	1000	64.1	Motor berputar	
3	2000	75.5	Motor berputar	
4	3000	112.5	Motor berputar.	
5	4000	157.7	Motor bergerak namun bergetar.	

6	5000	187.4	Motor bergerak namun bergetar.	
7	6000	-	Motor tidak bergerak hanya bergetar.	-

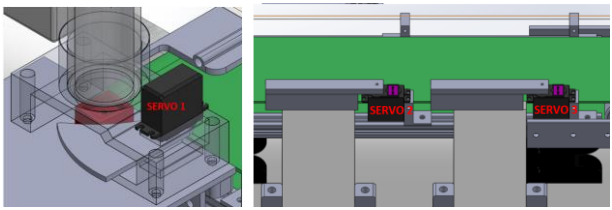
Hasil pengujian pada tabel dibawah menunjukan seberapa cepat *Motor Stepper* dalam nilai RPM dan mengetahui kondisi pada *Motor Stepper*. Disini dijelaskan bahwa *Motor Stepper* dapat berputar dengan pengaturan nilai frekuensi direntang 500hz-5000hz dan tidak dapat berputar jika pengaturan nilai frekuensi berada dikisaran 6000hz-8000hz, pada sistem pengendali motor yang memiliki frekuensi terlalu tinggi dapat menyebabkan kemacetan, karena sistem kendali motor mengirimkan terlalu banyak pulsa dalam waktu yang terlalu singkat. Dengan begitu pada alat untuk menggerakan *Motor Stepper* menggunakan 1000hz.

2. Pengendali *Servo*

Pada pengujian pengendali *Servo* menggunakan *pin input* dari arduino untuk menggerakan *Servo*. Proses pengujian ini dilakukan dengan mengatur nilai *Servo.write* pada *software* Arduino, nilai yang di berikan pada *Servo.write* dari 0 – 180 dikarenakan menggunakan *Servo* mg996 180°.

Tujuan dari pengujian pengendali *Servo* , dimana *Servo* 1 dapat mendorong objek benda dan *Servo* dapat menahan objek benda selanjutnya, *Servo* 2 dan *Servo* 3 dapat mengambil objek benda yang akan di pisahkan ketempat pemilahan.

Pada gambar 23 menunjukan posisi pada *Servo*, dimana pada posisi *Servo* dapat menentukan kondisi awal derajat pada *Servo*, dengan begitu dapat mengetahui *Servo* tersebut berputar ke arah kanan atau kiri dengan mengatur derajat pada *Servo.*



Gambar 13. Posisi pada *Servo*

Tabel 18. Kondisi *Servo*

No	Nama	Kondisi awal *Servo*(°)	Setting derajat *Servo*(°)	Kondisi *Servo*	Kondisi benda
1	*Servo* 1	180	25	Dapat mendorong objek benda	Objek benda tidak dapat keluar dari penempatan objek benda.
			50	Dapat mendorong objek benda	Objek benda hanya terlihat sedikit.
			75	Dapat mendorong objek benda	Tepat ke dalam area dan bisa menahan benda selanjutnya.
			100	Dapat mendorong objek benda	Tepat ke dalam area dan bisa menahan benda selanjutnya.
			125	Dapat mendorong objek benda	Objek benda melewati batas area dan tidak dapat menahan objek benda selanjutnya.
2	*Servo* 2	0	155	Dapat terbuka	Objek benda tidak dapat masuk dalam box pemilah.
			130	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
			105	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
			80	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
			55	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
3	*Servo* 3	0	155	Dapat terbuka	Objek benda tidak dapat masuk dalam box pemilah.

			130	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
			105	Dapat terbuka	Objek benda dapat masuk dalam box pemilah
			80	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.
			55	Dapat terbuka	Objek benda dapat masuk dalam box pemilah.

Dari hasil pengujian tabel 10 menunjukan bahwa dari *settingan* derajat *Servo* dapat menentukan kondisi *Servo* dan pengambilan benda objek. Pada *Servo* 1 *settingan* derajat 25-50° *Servo* dapat mendorong tetapi tidak sampai area deteksi, sedangkan *setting* derajat 70-90° maka *Servo* dapat mendorong objek benda dan menahan objek benda dan *settingan* derajat lebih dari 135° *Servo* dapat objek benda tetapi tidak bisa menahan objek benda selanjut nya.

Pada *Servo* 2 dan 3 *settingan* derajat dari 45–130° *Servo* dapat mengambil objek benda, pada derajat 155° *Servo tidak* dapat mengambil objek benda dengan baik. Pada alat derajat yang digunakan untuk menggerkan *Servo* 1 yakni 75° sedangkan pada *Servo* 2 dan 3 derajat yang digunakan 130°.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah dilakukan perancangan dan pengujian pada *Camera Based Smart Sorting on Production Line* ini, maka secara keseluruhan dapat diambil kesimpulan sebagai berikut :

1. Selama tahap pengujian objek, kamera berhasil mengenali atau mendeteksi setiap objek berdasarkan warna dan bentuk dengan akurasi dan presisi yang melebihi 80%. Pengujian ini membuktikan dalam mengatasi masalah dalam pemilahan manual dan meningkatkan efisiensi produksi.
2. Pada pengujian Komunikasi Modbus RTU (*Remote Terminal Unit)* yang digunakan mampu menghubungkan data PC ke PLC (*Programmable Logic Controller*) *Outseal*. Dalam hal ini PC berperan sebagai data *master* yang mengelurkan perintah untuk dikirimkan ke PLC (*Programmable Logic Controller*) *Outseal* sebagai *slave* yang akan menerima data, maka PLC (*Programmable Logic Controller*) *Outseal* akan melakukan tugasnya berdasarkan data yang diterima.
3. Pada pengujian sistem Kendali *Motor Stepper* pada alat ini menggunakan pengaturan 1000 frekuensi, dengan konfigurasi ini, *Motor Stepper* akan menggerakkan konveyor untuk memindahkan objek benda ke dalam *box* pemilahan. Selain itu, pada pengujian sistem kendali *Motor Servo* pada *Servo* 1, 2, dan 3, *Servo* dapat mendorong objek benda kedalam area dan memasukkan objek benda kedalam *box* pemilahan berdasarkan bentuk. Pengaturan *Servo* 1 pada rentang 70-100 derajat sementara *Servo* 2 dan 3 di atur pada 130 derajat. Maka terjadilah pemilahan pada setiap bentuk objek benda.

5.2. Saran

Dalam pengerjaan penelitian tugas akhir ini masih terdapat beberapa kekurangan yang perlu diperhatikan. Berdasarkan hal terebut, penulis dapat memberikan saran sebagai berikut:

1. Untuk peneliti selanjutnya diharapkan melakukan pengembangan lebih lanjut dengan menambahkan fungsionalitas baru sesuai dengan kebutuhan pengguna.
2. Untuk peneliti selanjutnya data melakukan penambahan box pemilahan pada setiap objek benda. Di karenakan pada penelitian ini hanya ada 3 *box* pemilhanan objek benda bulat, kotak dan bintang. Disarankan untuk menambah *box* pemilah menjadi 18 pemilah atau masing-masing warna dan bentuk.

3. Hasil penelitian berikutnya diharapkan untuk membuat banyak bentuk dan warna.

Daftar Pustaka

[1] F. R. Wicaksono, A. Rusdinar, I. Prasetya, and D. Wibawa, "PERANCANGAN DAN IMPLEMENTASI ALAT PENYORTIR BARANG PADA KONVEYOR DENGAN PENGOLAHAN CITRA DESIGN AND IMPLEMENTATION OF ITEMS DEVICE SORTING ON CONVEYOR WITH IMAGE PROCESSING," 2018.

[2] H. S. Risfendra, "Otomasi Industri Dengan Arduino *Outseal* PLC," UNP PRESS. Accessed: May 29, 2023. [Online]. Available: https://books.google.co.id/books/about/Otomasi_Industri_Dengan_Arduino_*Outseal*.html?id=kHAQEAAAQBAJ&redir_esc=y

[3] A. Al-Naji, A. B. Fakhri, S. K. Gharghan, and J. Chahl, "Soil color analysis based on a RGB camera and an artificial neural network towards smart irrigation: A pilot study," *Heliyon*, vol. 7, no. 1, Jan. 2021, doi: 10.1016/j.heliyon.2021.e06078.

[4] C. Chen, Z. Li, and L. Fu, "Perovskite photodetector-based single pixel color camera for artificial vision," *Light Sci Appl*, vol. 12, no. 1, p. 77, Mar. 2023, doi: 10.1038/s41377-023-01127-0.

[5] M. I. Sari, R. Fajar, T. Gunawan, and R. Handayani, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION The Use of Image Processing and Sensor in Tomato Sorting Machine by Color, Size, and Weight," 2022. [Online]. Available: www.joiv.org/index.php/joiv

[6] A. Suryowinoto and A. Zihar Wirandi, "Pengembangan Sistem Pemilah dan Pengelompokan Penghitung Objek Produksi pada Konveyor Berbasis Kamera dengan Metode RGB Threshold," Jun. 2021.

[7] A. Aprizaldi Dalimunthe, N. Ariyanto Adli, and D. Teknik Elektronika, "PROTOTIPE ROBOT LENGAN 3 DEGREE OF FREEDOM SEBAGAI ALAT SORTING BARANG BERDASARKAN WARNA BARANG BERBASIS INTERNET OF THINGS," 2019, doi: 10.21009/autocracy.06.2.2.

[8] N. Arifin, D. M. Sari, and A. Chairy, "Prototype Pemilah Buah Stroberi Otomatis menggunakan Kamera berbasis Arduino Uno," *JCIS (Journal of Computer and Information System)*, vol. 4, no. 2, pp. 42–50, 2021, doi: 10.31605/jcis.v4i2.

[9] A. Fahmi Fandisyah, N. Iriawan, and W. Setya Winahju, "Deteksi Kapal di Laut Indonesia Menggunakan," 2021.

[10] M. Sarosa and N. Muna, “IMPLEMENTASI ALGORITMA YOU ONLY LOOK ONCE (YOLO) UNTUK DETEKSI KORBAN BENCANA ALAM,” vol. 8, no. 4, 2021, doi: 10.25126/jtiik.202184407.

[11] K. S. Nugroho, “Confusion Matrix untuk Evaluasi Model pada Supervised Learning,” https://ksnugroho.medium.com/. Accessed: Jun. 08, 2023. [Online]. Available: https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

[12] A. Habib Patonra, S. Masita, N. R. Wibowo, A. Fitriati, and P. Bosowa, “Rancang Bangun Media Pembelajaran Praktik Motor Stepper,” 2020.

[13] B. Cahyo Wibowo and F. Nugraha, “Stepper Motor Speed Control Using Start-Stop,” 2021.

[14] “Driver Motor Stepper: Jenis dan Aplikasinya.” Accessed: May 29, 2023. [Online]. Available: https://abdulelektro.blogspot.com/2021/04/driver-motor-stepper-jenis-dan.html

[15] M. R. G. A. P. S. K. Mulyadi Indra H, “Modul Komunikasi Modbus RTU over RS485 Berbasis Arduino,” 2021.

[16] S. Rachman, “KOMUNIKASI ANTARMUKA PROGAMABLE LOGIC CONTROLLER PADA MODBUS RTU SENSOR SUHU DAN KELEMBAPAN UDARA DENGAN DATA LOGGER,” POLITEKNIK NEGERI BANJARMASIN. Accessed: May 29, 2023. [Online]. Available: https://www.researchgate.net/publication/369119634_KOMUNIKASI_ANTARMUKA_PROGRAMABLE_LOGIC_CONTROLLER_PADA_MODBUS_RTU_SENSOR_SUHU_DAN_KELEMBABAN_UDARA_DENGAN_DATA_LOGGER

[17] O. Saputra, “Komunikasi *Outseal* Plc dengan Smartphone,” 2022, doi: 10.38035/rrj.v4i4.

[18] D. A. Pangestu *et al.*, “Tempat Sampah Otomatis Menggunakan Kendali Loop Terbuka,” pp. 26–27, 2020.

[19] I. H Mulyadi, R. Mahdaliza, A. Gautama, S. Prayoga, and Kamarudin, “Modul Komunikasi Modbus RTU over RS485 Berbasis Arduino,” Jul. 2021, doi: 10.1002/aic.10279.

Biodata

Nama : Aditya Muhammad Kinandhika
TTL : Bandar Lampung, 13 Maret 1999
Agama : Islam
Alamat : Kav. Flamboyen Blok L No 57

Email : Adityakinandhika@gmail.com
Riwayat Pendidikan SMA/SMK : SMK Negeri 001 Batam
SMP : SMP Negeri 27 Batam

Nama : Putri Lidiya
TTL : Batam, 10 Juni 2001
Agama : Islam
Alamat : Kampung Harapan Swadaya Blok AG 6 No 10
Email : Putri.lidya1001@gmail.com
Riwayat Pendidikan SMA/SMK : SMA Negeri 8 Batam
SMP : SMP Negeri 6 Batam

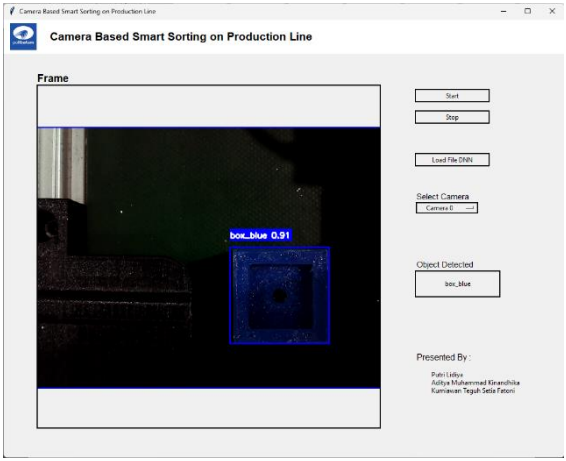
Nama : Kurniawan Teguh Setia Fatoni
TTL : Malang, 16 Juni 2000
Agama : Islam
Alamat : Perumahan GMP Blok N No 87

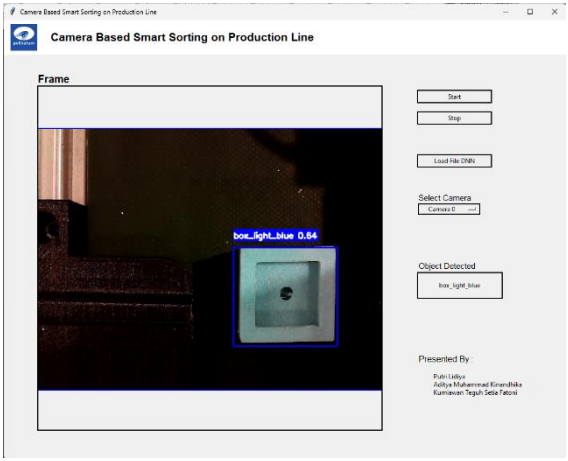
Email : ttkurniawan4@gmail.com
Riwayat Pendidikan SMA/SMK : SMK Negeri 3 Batam
SMP : SMP Swasta Laksamana Batam

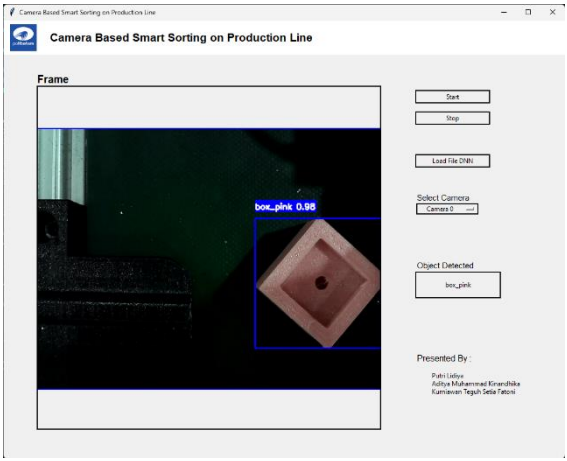
Lampiran

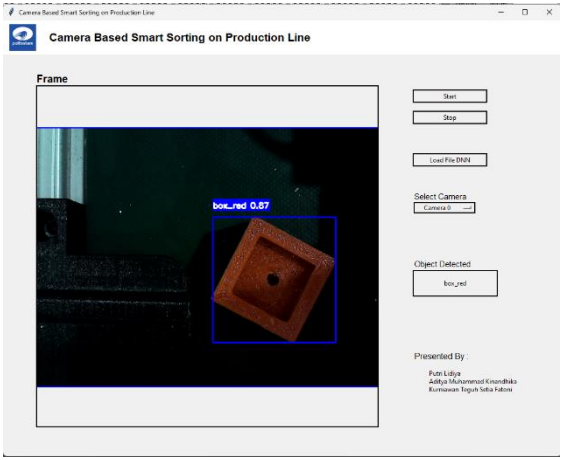
Lampiran A (Bentuk alat)

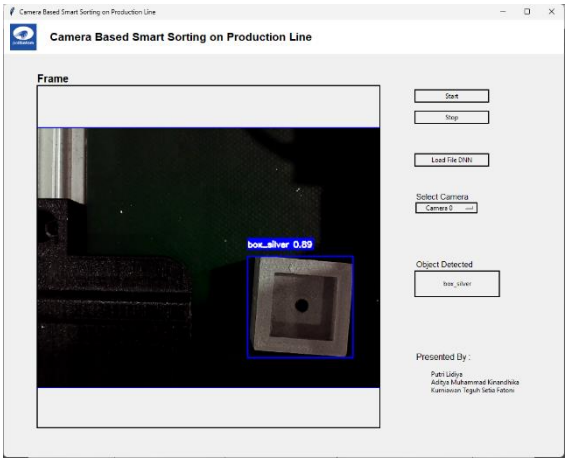
Lampiran B (Deteksi GUI (*Graphical User Interface*))

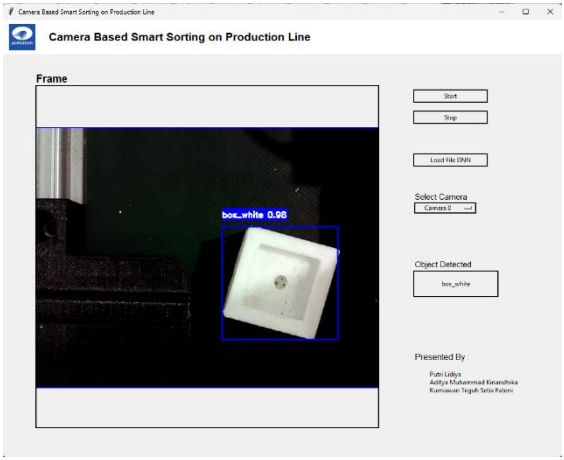


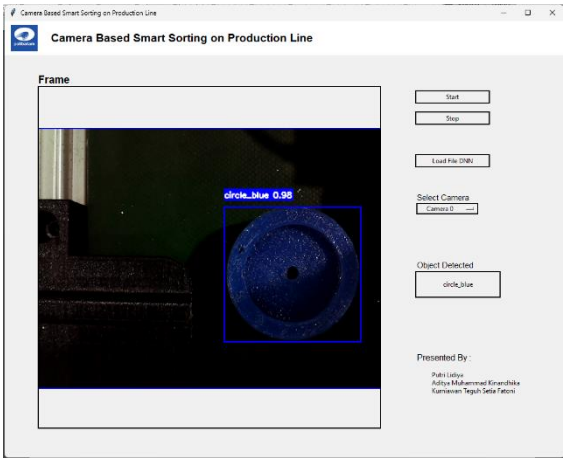


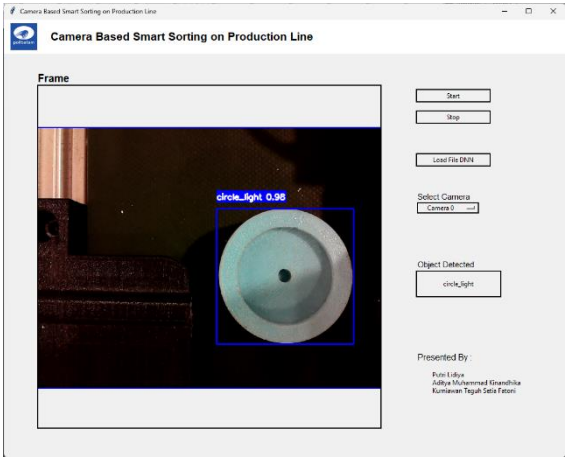

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
box_pink 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
box_pink
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

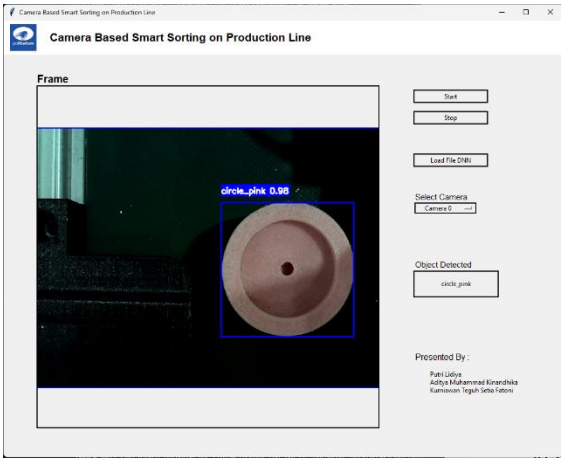

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
box_red 0.87
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
box_red
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

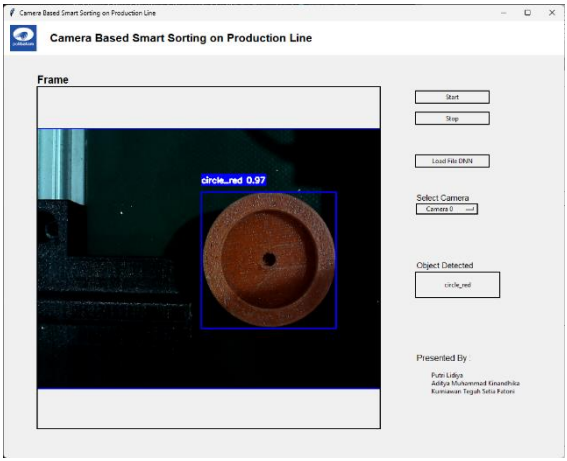

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
box_silver 0.89
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
box_silver
Presented By :
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Putoni

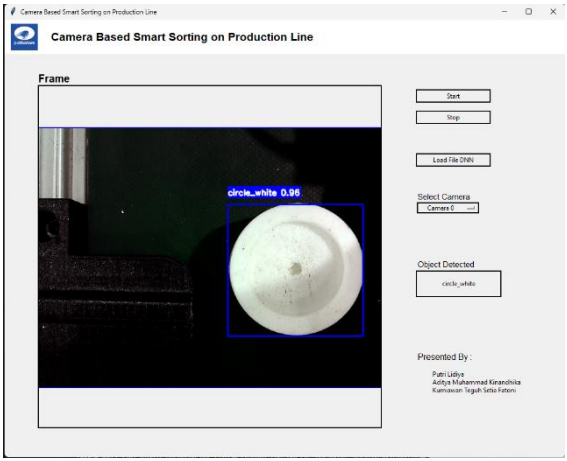

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
box_white 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
box_white
Presented By :
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Putoni

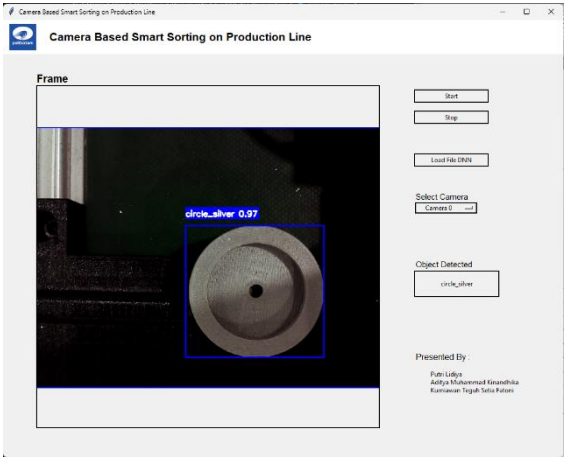

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_blue 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_blue
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

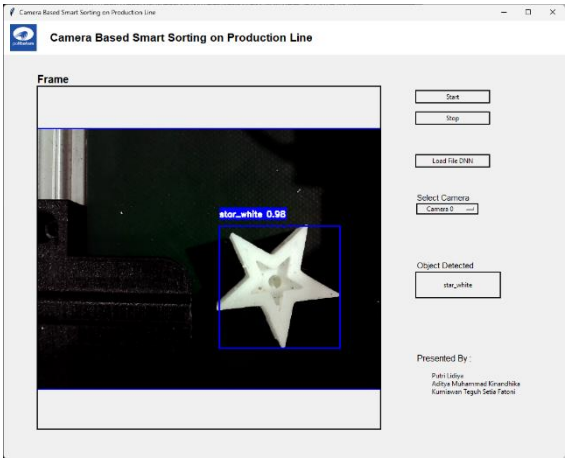

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_light 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_light
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

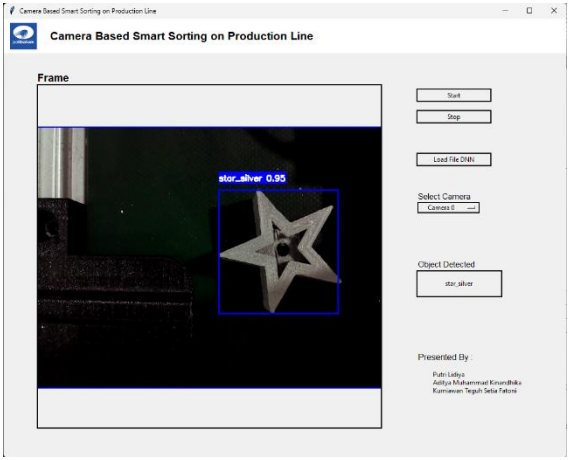

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_pink 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_pink
Presented By :
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

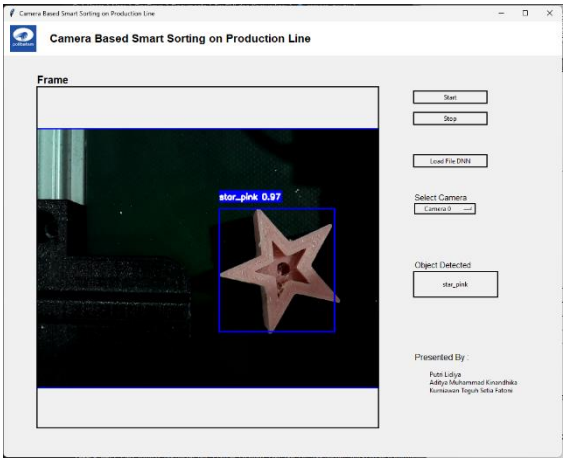

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_red 0.97
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_red
Presented By :
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

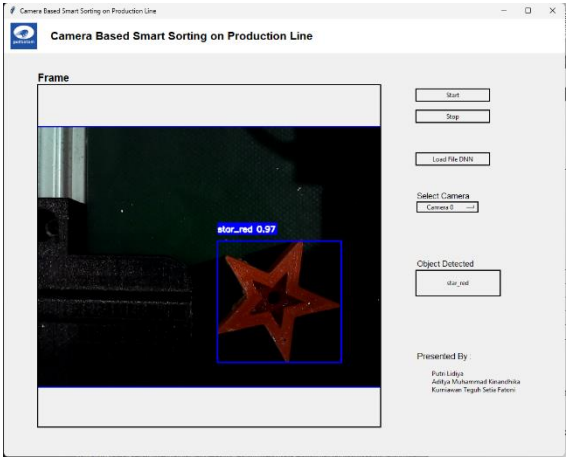

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_white 0.96
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_white
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

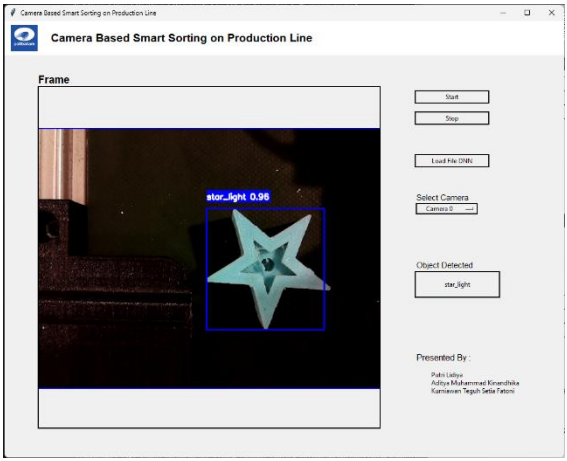

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
circle_silver 0.97
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
circle_silver
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

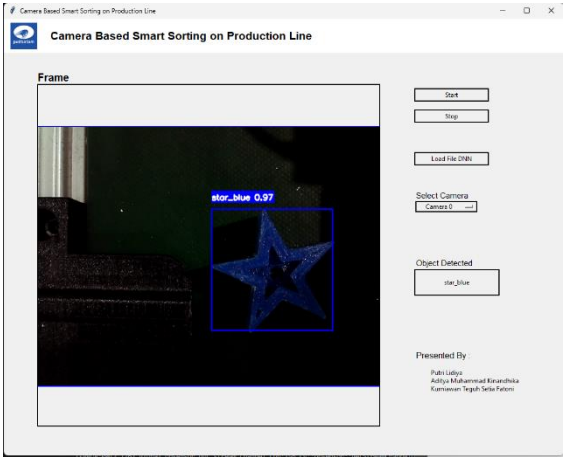

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_white 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
star_white
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni


Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_silver 0.95
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
star_silver
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

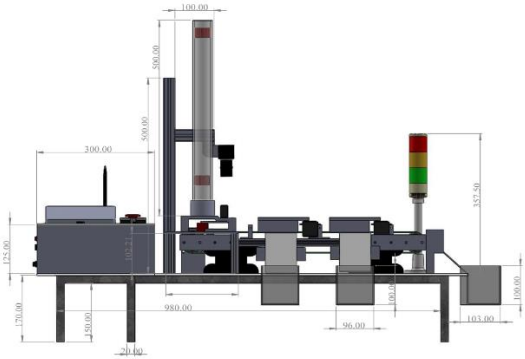

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_pink 0.97
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
stor_pink
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

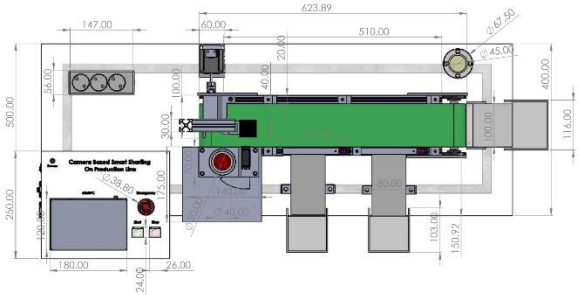

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_red 0.97
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
stor_red
Presented By :
Putri Lidya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

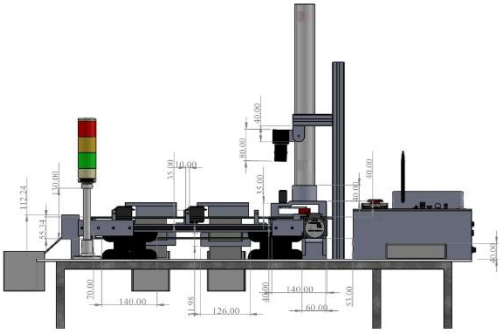

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_light 0.98
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
star_light
Presented By :
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

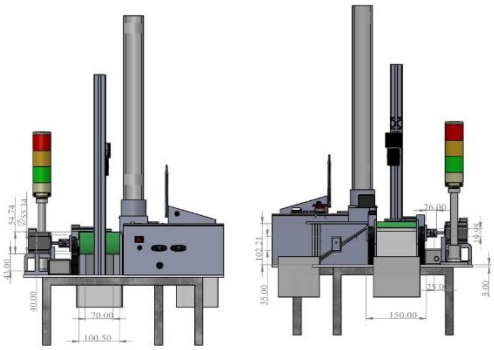

Camera Based Smart Sorting on Production Line
Camera Based Smart Sorting on Production Line
Frame
stor_blue 0.97
Start
Stop
Load File DNN
Select Camera
Camera 0
Object Detected
star_blue
Presented By
Putri Lidiya
Aditya Muhammad Kinandhika
Kurniawan Teguh Setia Fatoni

Lampiran C (ukuran Alat)

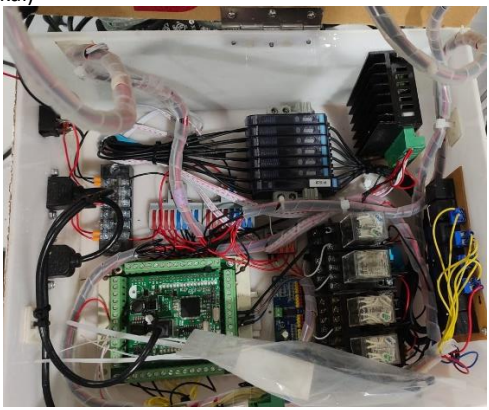


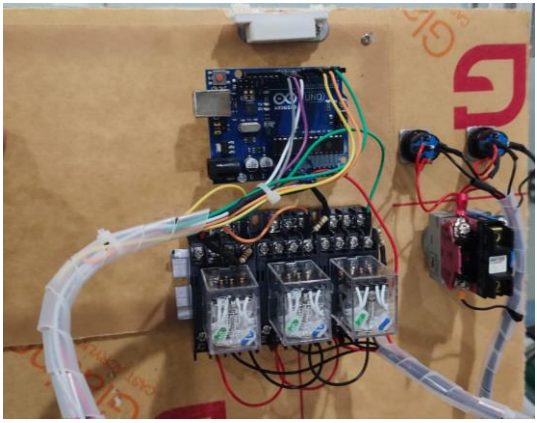




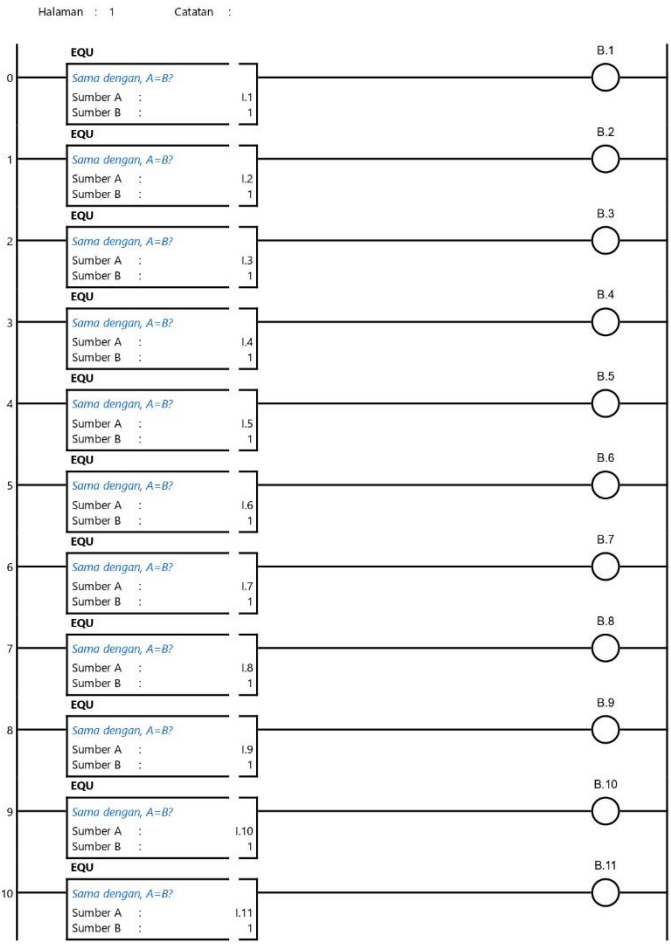

54.74
Ø55.34
16.00
80.00
70.00
100.50
102.21
35.00
26.00
17.00
150.00
5.00

Lampiran E (eletrikal)

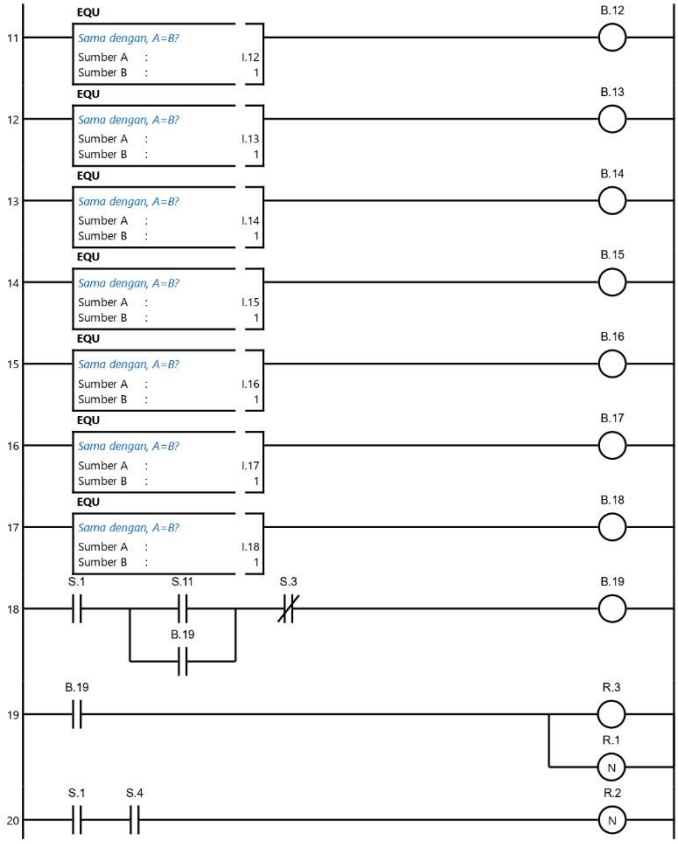




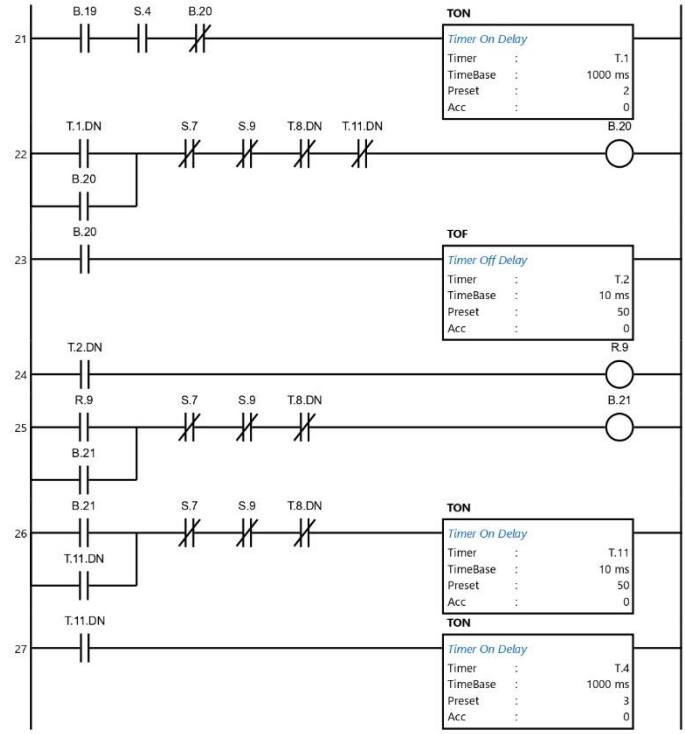
Lampiran F (Program PLC (*Programmable Logic Controller*) *Outseal*)



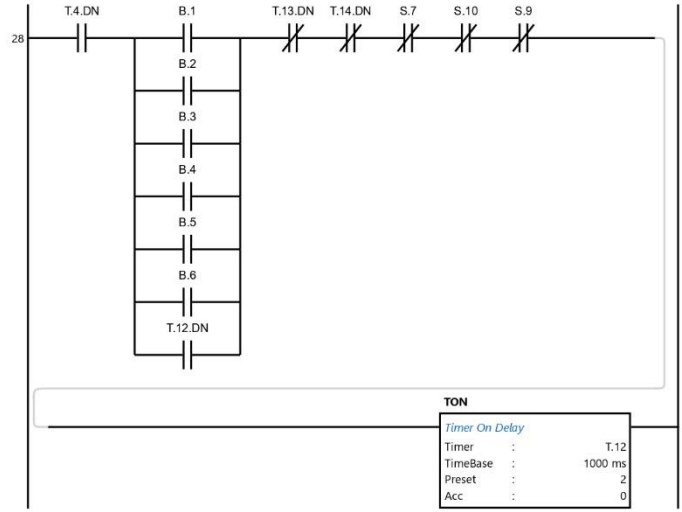
Halaman : 2 Catatan :



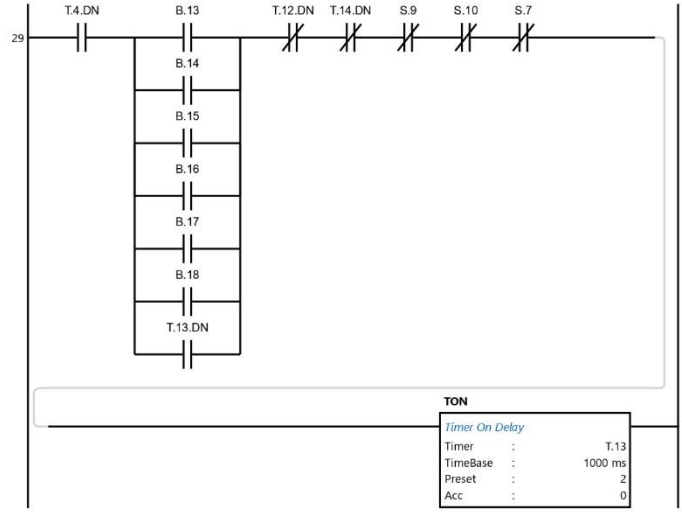
Halaman : 3 Catatan :



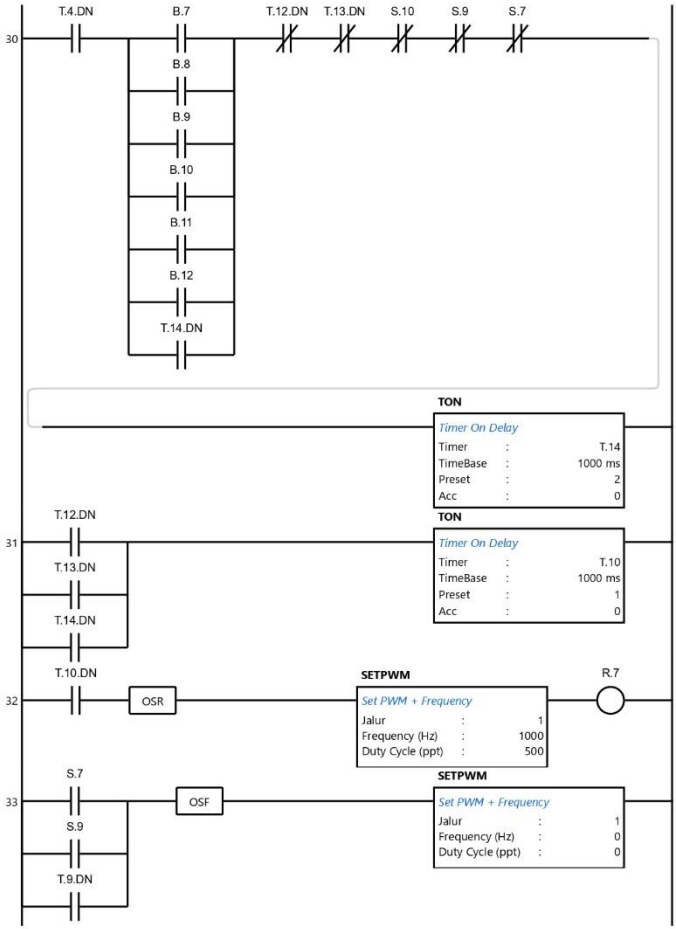
Halaman : 4 Catatan :



Halaman : 5 Catatan :



Halaman : 6 Catatan :



Lampiran G (Program modbus dan GUI (*Graphical User Interface*))

```
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 = 0.4
NMS_THRESHOLD = 0.4


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("Camera based smart")
        width = 1050
        height = 820

        # Declare PLC
        self.instrument = minimalmodbus.Instrument('COM3', 1)
        self.instrument.serial.baudrate = 115200

        # Set the window title
        root.title("Camera Based Smart Sorting on Production Line")

        # Load the logo image
        logo_image = Image.open("logo_poltek.jpg")
```

```
        logo_image = logo_image.resize((50, 50), Image.Resampling.LANCZOS)
        self.logo_photo = ImageTk.PhotoImage(logo_image)

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

        # Create a label for the logo image
        logo_label = tk.Label(headline_frame, image=self.logo_photo, bg="white")
        logo_label.pack(side=tk.LEFT, padx=10, pady=5)

        # Create a label for the headline
        headline_label = tk.Label(headline_frame, text="Camera Based Smart Sorting
on Production Line", font=("Arial", 16, "bold"), bg="white")
        headline_label.pack(side=tk.LEFT, padx=10, pady=5)


        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 = ['box_blue', 'box_light_blue', 'box_pink', 'box_red',
'box_silver', 'box_white',
                'circle_blue', 'circle_light', 'circle_pink', 'circle_red',
'circle_silver', 'circle_white',
                'star_blue', 'star_light', 'star_pink', 'star_red', 'star_silver',
'star_white']

        self.class_to_register = {'box_blue': 0, 'box_light_blue': 1, 'box_pink':
2, 'box_red': 3, 'box_silver': 4, 'box_white': 5,
                    'circle_blue': 6, 'circle_light': 7, 'circle_pink': 8,
'circle_red': 9, 'circle_silver': 10, 'circle_white': 11,
                    'star_blue': 12, 'star_light': 13, 'star_pink': 14,
'star_red': 15, 'star_silver': 16, 'star_white': 17}

        self.default_register_value = 0


        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")

        camera_info = tk.Label(root, text="Select Camera", font=("Arial", 11))
        camera_info.place(x=770, y=320)
        GListBox_128 = tk.OptionMenu(root, self.selected_camera, *self.cameras)
        GListBox_128.config(bd=2, relief="solid")
```

```
        GListBox_128.place(x=770, y=340, width=120, height=25)

        frame_info = tk.Label(root, text="Frame", font=("Arial", 15, "bold"))
        frame_info.place(x=60, y=95)
        self.canvas = tk.Canvas(root, width=640, height=640, bd=2, relief="solid")
        self.canvas.place(x=60, y=120)

        GButton_536 = tk.Button(root, text="Start", command=self.turn_on_camera,
bd=2, relief="solid")
        GButton_536.place(x=770, y=130, width=140, height=25)

        GButton_599 = tk.Button(root, text="Stop", command=self.turn_off_camera,
bd=2, relief="solid")
        GButton_599.place(x=770, y=170, width=140, height=25)

        GButton_load = tk.Button(root, text="Load File DNN",
command=self.load_file, bd=2, relief="solid")
        GButton_load.place(x=770, y=250, width=140, height=25)

        terdeteksi_info = tk.Label(root, text="Object Detected", font=("Arial",
11))
        terdeteksi_info.place(x=770, y=447)
        self.detection_label = tk.Label(root, text="Print Data", bd=2,
relief="solid")
        self.detection_label.place(x=770, y=470, width=160, height=50)

        nama_info = tk.Label(root, text="Presented By :", font=("Arial", 12),
justify=tk.LEFT)
        nama_info.place(x=770, y=620)
        names = "Putri Lidiya\nAditya Muhammad Kinandhika\nKurniawan Teguh Setia
Fatoni"
        self.GLabel_368 = tk.Label(root, text=names, justify=tk.LEFT)
        self.GLabel_368.place(x=770, y=650, width=228, height=60)


    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")
            self.detection_label.config(text="")
```

```
    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()  # Start updating the canvas

    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)
                # frame = cv2.convertScaleAbs(frame, alpha=1.2, beta=30)
                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)
                detected_objects = [detection[4] for detection in detections]
                self.detection_label.config(text="\n".join(detected_objects))

                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}"

                    # Calculate the position to place the text above the bounding
box
                    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

                    # Draw the text background
                    cv2.rectangle(frame, (text_x, text_y - text_height - 5),
(text_x + text_width, text_y + 5), (0, 0, 255), -1)

                    # Draw the text
```

```
                    cv2.putText(frame, text, (text_x, text_y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

                # Get the size of the canvas
                canvas_width = self.canvas.winfo_width()
                canvas_height = self.canvas.winfo_height()

                # Resize the frame to fit the canvas while maintaining the aspect
ratio
                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))

                # Add border line to the frame
                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)

                for detection in detections:
                    try:
                        x1, y1, width, height, detected_class, confidence =
detection
                        if detected_class == 'box_blue':
                            self.instrument.write_register(0, 1, 0, 6)
                            self.instrument.write_register(1, 0, 0, 6)
                            self.instrument.write_register(2, 0, 0, 6)
                            self.instrument.write_register(3, 0, 0, 6)
                            self.instrument.write_register(4, 0, 0, 6)
                            self.instrument.write_register(5, 0, 0, 6)
                            self.instrument.write_register(6, 0, 0, 6)
                            self.instrument.write_register(7, 0, 0, 6)
                            self.instrument.write_register(8, 0, 0, 6)
                            self.instrument.write_register(9, 0, 0, 6)
                            self.instrument.write_register(10, 0, 0, 6)
                            self.instrument.write_register(11, 0, 0, 6)
                            self.instrument.write_register(12, 0, 0, 6)
                            self.instrument.write_register(13, 0, 0, 6)
                            self.instrument.write_register(14, 0, 0, 6)
                            self.instrument.write_register(15, 0, 0, 6)
                            self.instrument.write_register(16, 0, 0, 6)
```

```
            self.instrument.write_register(17, 0, 0, 6)
            #print("box blue")

        elif detected_class == 'box_light_blue':
            self.instrument.write_register(1, 1, 0, 6)
            for i in range(18):
                if i != 1:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("box light blue")

        elif detected_class == 'box_pink':
            self.instrument.write_register(2, 1, 0, 6)
            for i in range(18):
                if i != 2:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("box pink")

        elif detected_class == 'box_red':
            self.instrument.write_register(3, 1, 0, 6)
            for i in range(18):
                if i != 3:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("box red")

        elif detected_class == 'box_silver':
            self.instrument.write_register(4, 1, 0, 6)
            for i in range(18):
                if i != 4:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("box silver")

        elif detected_class == 'box_white':
            self.instrument.write_register(5, 1, 0, 6)
            for i in range(18):
                if i != 5:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("box white")

        elif detected_class == 'circle_blue':
            self.instrument.write_register(6, 1, 0, 6)
            for i in range(18):
                if i != 6:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("circle blue")

        elif detected_class == 'circle_light':
            self.instrument.write_register(7, 1, 0, 6)
            for i in range(18):
                if i != 7:
                    self.instrument.write_register(i, 0, 0, 6)
            #print("circle light")
```

```
elif detected_class == 'circle_pink':
    self.instrument.write_register(8, 1, 0, 6)
    for i in range(18):
        if i != 8:
            self.instrument.write_register(i, 0, 0, 6)
    #print("circle pink")

elif detected_class == 'circle_red':
    self.instrument.write_register(9, 1, 0, 6)
    for i in range(18):
        if i != 9:
            self.instrument.write_register(i, 0, 0, 6)
    #print("circle red")

elif detected_class == 'circle_silver':
    self.instrument.write_register(10, 1, 0, 6)
    for i in range(18):
        if i != 10:
            self.instrument.write_register(i, 0, 0, 6)
    #print("circle silver")

elif detected_class == 'circle_white':
    self.instrument.write_register(11, 1, 0, 6)
    for i in range(18):
        if i != 11:
            self.instrument.write_register(i, 0, 0, 6)
    #print("circle white")

elif detected_class == 'star_blue':
    self.instrument.write_register(12, 1, 0, 6)
    for i in range(18):
        if i != 12:
            self.instrument.write_register(i, 0, 0, 6)
    #print("star blue")

elif detected_class == 'star_light':
    self.instrument.write_register(13, 1, 0, 6)
    for i in range(18):
        if i != 13:
            self.instrument.write_register(i, 0, 0, 6)
    #print("star light")

elif detected_class == 'star_pink':
    self.instrument.write_register(14, 1, 0, 6)
    for i in range(18):
        if i != 14:
            self.instrument.write_register(i, 0, 0, 6)
    #print("star pink")

elif detected_class == 'star_red':
```

```
                            self.instrument.write_register(15, 1, 0, 6)
                            for i in range(18):
                                if i != 15:
                                    self.instrument.write_register(i, 0, 0, 6)
                            #print("star red")

                        elif detected_class == 'star_silver':
                            self.instrument.write_register(16, 1, 0, 6)
                            for i in range(18):
                                if i != 16:
                                    self.instrument.write_register(i, 0, 0, 6)
                            #print("star silver")

                        elif detected_class == 'star_white':
                            self.instrument.write_register(17, 1, 0, 6)
                            for i in range(18):
                                if i != 17:
                                    self.instrument.write_register(i, 0, 0, 6)
                            #print("star white")

                        else:
                            for i in range(18):
                                self.instrument.write_register(i, 0, 0, 6)
                            pass

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

                self.root.after(10, self.update)  # Schedule the next 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):
        # Implement NMS
        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 Arduino)

```
#include <Servo.h>
int tombol = 3;
int tombol2 = 4;
int tombol3 = 5;
int buttonstate = 0;
int buttonstate2 = 0;
int buttonstate3 = 0;

Servo servo;
Servo servo2;
Servo servo3;

void setup() {
  servo.attach(9);
  servo2.attach(10);
  servo3.attach(11);
  servo.write(180);
  servo2.write(0);
  servo3.write(0);

  pinMode(tombol, INPUT);
  pinMode(tombol2, INPUT);
  pinMode(tombol3, INPUT);
}
void loop() {
  buttonstate = digitalRead (tombol);
  buttonstate2 = digitalRead (tombol2);
  buttonstate3 = digitalRead (tombol3);
  if (buttonstate == 1){
    servo.write(0);
  }
  else {
    servo.write(75);
  }
  if (buttonstate2 == 1){
    servo2.write(180);
  }
  else {
    servo2.write(130);
  }
  if (buttonstate3 == 1){
    servo3.write(180);
  }
  else {
    servo3.write(130);
  }
}
```